AN ASHTON-TATE SOFTWARE SUPPORT PUBLICATION OCTOBER 1986

# TechNotes

OCT 10 1986

Ashton-Tate Announces:
- No Copy Protection
- Custom Support

dBASE III PLUS Change Summary –Part Four

dBASE Picture Clause for Framework II

ASHTON-TATE

# TechNotes

TABLE OF CONTENTS

## FEATURES


3 Ashton-Tate Announces "Custom Support"
*by Laurie Miller*

9 No More Copy Protection

12 dBASE III PLUS Change Summary—Part Four
*by Joseph Perret*

21 Simulating the dBASE Picture Clause for Data Input
*by Karen Robinson and Kent Irwin*

26 Optimizing Network Techniques
*by J. Mike Hedblom*


## DEPARTMENTS


1 Read Me First
*by Laurie Miller*

36 Usage Tips

44 Anomalies

48 Corrigenda

49 On-Line Services

54 MultiMate Technical Support

59 International Support

62 Letters

# READ ME FIRST

*by Laurie Miller*

As I write this column, it is early August and few people here at Ashton-Tate know that, by the end of this month, we will have announced new versions of our software that feature *NO COPY PROTECTION*. It's similar, I suppose, to catching a glimpse of the future, knowing what will happen but only guessing at what the reaction will be.

Most of you probably have, at one time or another, cursed the very idea of copy-protected software. So, no doubt, you are as happy as I am to know that all the "I hate copy protection" mail you sent to us, and all those phone calls you placed to Software Support to voice your agony over trying to install a SUPERLoK'd product to your hard disk—it has all been heard and it has finally paid off.

For information on these new non-copy-protected products, Framework II, dBASE III PLUS, and dBASE III PLUS LAN PACK versions 1.1, read "No More Copy Protection!" This article overviews the new versions and, in particular, reviews the new installation procedures. For further technical information, refer to part 4 of the dBASE III PLUS Change Summary, also in this issue. The last in our series, this reports some of the anomalies that existed in dBASE III and/or the Developer's Release were fixed in PLUS, version 1.0. Currently, we are assembling the dBASE III PLUS version 1.1 Change Summary, which will report all changes made to this version of the software from version 1.0. Look for this in the November issue.

Personally, I couldn't be happier for our users. In a few short weeks, when the news is announced, it'll be party time in tech support! And for another good reason.....

When no copy protection was announced, so was the Custom Support Plan. This new support services program has been undergoing development ever since I started with Ashton-Tate, back in August 1984. And it's finally a reality! We at the Software Support Center are pleased that we are now in a better position to give the high-quality support you've been demanding.

The article "Ashton-Tate Announces Custom Support" gives you an overview of this new service: what it is, how it works, and why we decided to do it. Included is a set of questions and answers, which will answer some of the questions you most likely have about the Plan. We understand that paying for phone support is a new, and perhaps awkward, step for our customers to take. We are sensitive to your concerns about paying for this service, and so it is important for us to address them in ***TechNotes***.

I hope you enjoy these and the other articles and technical info we're offering you in this October issue!

TechNotes

**Publisher**
*Ashton-Tate Software Support Center*
*Jay Hanson, Director*
*Roy Moore, Manager*

**Editor-in-Chief**
*Laurie Miller*

**Editors**
*Christopher White*

**Technical Writers**
*Brenda Johnson-Grau*
*Karen Robinson*

**Assistant Editor**
*Olivier Biggerstaff*

**Technical Staff**
*Larry Abare, Richard Augelli, Stanley Ballenger, Rodney Barbati, Jeffrey Bersin, Richard Bunter, Mike Carlisle, Ray Cortenbach, Fred Crownover, Michael Fiore, Rick Fillman, Charles Gehman, Ken Getz, Richard Goepel, Peter Gold, Mina Gussin, Doug Haman, Scott Haynes, Mike Hedblom, Clint Helton, Jack Hirsch, David Howlett, Errol Jackson, Matt Johnson, Rosaline Keenan, Gail Kulikowsky, Steven Kurasch, Jerry Levy, Chris Lo Vasco, Jeffrey McCrimon Tom Mangelsdorf, Jerry Mendes, Vincent Mendillo, Elizabeth Miranda, Jeffrey Moore, Keith Mund, Joseph Perret, Jaspal Rattan, John Regen, Deborah Ridolfo, Daryl Saybold, Steve Silverwood, Joseph Stoltz, Naoma Stow, Ralph Wakefield, Lisa Wanek, Quinn Wildman, Roger Wegehoft, Kari Wolff*

**Contributing Editors**
*Paul Bukauskas (International),*
*Liz Burke (MultiMate),*
*Steve Silverwood (On-Line Services)*

**Administrative Assistant**
*Kathy Smith*

**Illustrator**
*Stephen Goodman*

**Production Staff**
*D. A. Gray Bare, Tim Dunne, Jeff Elliott, Darrell Evers, Louis Hirsch, R. James Landers, Renée Maynard, Patty Pesek,*

**Product Development**
*Forefront Development Center:*
*Marilyn McMahon, Greg Stikeleather*
*Glendale Development Center:*
*Debby Moody*

# DISCLAIMER NOTICE

The information contained in this publication is designed to help you develop and improve your programming skills and to assist you in using Ashton-Tate products. Ashton-Tate assumes no responsibility whatsoever for the uses made of any of the software code in this issue, whether modified or unmodified.

If you discover what you believe is an anomaly (as defined under **Anomalies**) not presented in this issue or others, please send a detailed description of the possible anomaly to:

Ashton-Tate Software Support Center
20101 Hamilton Ave.
Torrance, CA 90502-1319

If the reported anomaly can be reproduced by our technicians, it may be added to a future issue of ***TechNotes***.

The policies that are described in ***TechNotes*** apply only to the United States (and possessions), Canada, and Mexico. Residents of other foreign countries should contact a local Ashton-Tate representative in their country for policy information and technical support.

Ashton-Tate and the authors will not be liable for special, incidental, consequential, indirect or other similar damages, even if we have been advised of the possibility of such damages. This means we are not responsible for damages or costs incurred as a result of loss of time, loss of data, loss of profits or revenues, or loss of use of the software. In addition, we are not responsible for damages or costs incurred in connection with obtaining substitute software, claims by others, inconvenience or similar costs.

No warranties, either expressed or implied, are made regarding the contents of this publication, its merchantability or fitness for any particular purpose, all of which are hereby expressly disclaimed. Note, however, that some states do not allow the exclusion or limitation of warranties or of direct, incidental or consequential damages, so these warranty limitations may not apply to you.

In no event will our liability for any damages to you or any other person ever exceed the price paid for your subscription to ***TechNotes***, regardless of any form of the claim.

Additional statements by agents, employees, distributors and dealers of Ashton-Tate do not constitute warranties by Ashton-Tate and do not bind Ashton-Tate.

---

***TechNotes*** (ISSN 0883-6493) is an Ashton-Tate Software Support Center publication containing technical reference information on Ashton-Tate and MultiMate products.

Entire contents Copyright © Ashton-Tate 1986. All rights reserved. Commercial reproduction of the material appearing in ***TechNotes*** is forbidden without prior written permission. Direct all requests for reproduction permission to Ashton-Tate Software Support Center, 20101 Hamilton Ave., Torrance, CA 90502-1319 ATTN: Laurie Miller.

***TechNotes*** is published monthly at a subscription price of $50.00 for one year. Additional rates: Canada and Mexico, $65.00; other countries, $75.00. Monthly issues are available in back issues for $10.00 each (1985 issues) and $7.00 each (1986 issues). ***TechNotes*** is published by the Ashton-Tate Software Support Center, 20101 Hamilton Ave., Torrance, CA 90502-1319. POSTMASTER: Send subscription inquiries and address changes to ***TechNotes*** Subscription Service, P.O. Box 3729, Escondido, CA 92025-0929.

dBASE II, dBASE III PLUS, and Framework II are registered trademarks of Ashton-Tate. dBASE, RunTime, FRED, and ASHTON-TATE are trademarks of Ashton-Tate. Some of the trademarks of other companies mentioned in this issue appear for identification purposes only.

The Ashton-Tate Software Support Center's hours are 6:00 a.m. to 4:30 p.m., PST, Monday through Friday, except on Thursday when we close at 4:00 p.m. To reach our technicians for technical support, call (213) 329-0086. Be sure to have your software serial number ready for the technician.

# Ashton-Tate Announces "Custom Support"

*by Laurie Miller*

In August, Ashton-Tate announced several new services. The most prominent of these is the Custom Support Plan, an expanded software service and support program for all Ashton-Tate registered users. Under the Custom Support Plan, you must purchase a "support package" after an initial 90-day period, if you wish to receive direct access phone support from Ashton-Tate.

Whether you have heard about Custom Support or not, you probably have many questions about it. In this article, I have tried to anticipate some of your questions and concerns, and to address them for you.

## Why Should I Pay for Support?

As Ashton-Tate products have increased and expanded in number and complexity, so have the costs of supporting them. Rather than pass these costs on to you by increasing the price of our products, we felt it would be best if we offered support as an extended service to those users who want it. In addition, the revenue generated from our Custom Support Plan will be used primarily to offset these higher support costs so that we can provide more efficient, professional-quality service to our subscribers. In short, our primary goal is to continue to meet your ever-changing support needs by offering the best technical support in the microcomputer software industry.

## What is the Custom Support Plan?

Custom Support was developed to satisfy your support needs at any level, from novice to expert. Custom Support exists to provide the highest level technical support by customizing support services to individual needs. Over the last year, Ashton-Tate formulated, tested, and refined each element of the Custom Support Plan until it received a consistently positive response from market surveys and focus groups.

## How Custom Support Works

Effective September 1, 1986, all existing Ashton-Tate users, both registered and unregistered, began receiving 90 days of free, unlimited, direct access phone support. After September 1, the 90-day period began on the date that you purchased your software, **provided that you return your registration card to us within 30 days of purchase.** If you elect not to send in your software registration card within the 30-day grace period, you forfeit this "warranty support."

After your 90-day warranty support period expires, you must purchase one of three support packages to continue receiving direct access telephone support from Ashton-Tate. Each Custom Support package offers an increasing level of support: Extended Support (two levels) and Professional Support. Details about each package are fully described below.

## Extended Support Packages

The following outlines the two Extended Support Packages that are available. With either of them, you are purchasing a number of direct access phone support calls that you can make to the Software Support Center.

*Extended Support Package I*

- 1 year (maximum) direct access phone support
- **10** support calls
- 30% off Ashton-Tate books
- Cost is $50.00

*Extended Support Package II*

- 1 year (maximum) direct access phone support
- **20** support calls
- 30% off Ashton-Tate books
- Cost is $80.00

If you wish to, you may also buy one or more of the following "a la carte" products, both of which are included in the Professional Support Package, described below:

> ***TechNotes*** - one-year subscription (12 monthly issues): $50, domestic; $65, Canada and Mexico; $75, other foreign countries
>
> **CompuServe** - a consumer subscription: $30.00

## Professional Support Package

Professional Support is designed for the professional or "power" users of our software. If you use Ashton-Tate products extensively and need fast answers to your high-level support questions, then we suggest you purchase the Professional Support Package. With the Professional Support Package, you receive the following services and products:

*Professional Support Package*

- 1 year (maximum) direct access phone support via our *Express Support Queue*
- 20 support calls
- Access to an Ashton-Tate electronic mailbox
- One-year subscription to ***TechNotes*** (12 issues)
- Subscription to CompuServe Information Network
- The Software Support Center's Utility Disk

- Access to *Close-Up* (by Norton-Lambert), sessions are billed at $75.00 per hour
- 30% off Ashton-Tate books
- Cost is $150.00

**Express Support Queue**

The *Express Support Queue* is your assurance that you will speak with one of our most knowledgeable, experienced technicians as quickly as possible. Because fewer calls pass through this queue and manpower allocated to it is proportionately higher than our other queues, we can deliver rapid response. If your call is not answered within our targeted response time, an attendant will remove you from the queue and take a message for a return call at your option. Return calls are usually made the same day.

***TechNotes***

***TechNotes*** features the hottest technical information available in print on Ashton-Tate and MultiMate products. Every month the ***TechNotes*** editorial staff brings you timely support material in technical articles, including sample programs written by our technicians, and these regular monthly departments:

- **Usage Tips**, including quickie macros, user-defined functions, and handy tricks
- **Anomalies**, known software problems with work-arounds
- **On-Line Services**, excerpts from Ashton-Tate's SIG on CompuServe
- **International Support**
- **MultiMate Technical Support**
- **Letters**

Since ***TechNotes*** began in February 1985, our goal has been to serve as an authoritative and useful source of information on the current state of Ashton-Tate products. As part of the new Custom Support Plan, we will continue to uphold this standard.

**CompuServe**

A subscription to CompuServe is included in the Professional Support Package, along with $15 worth of free connect time. You can access Ashton-Tate technical information directly from our CompuServe technical database and forum. This information is updated weekly. You may also participate in the Ashton-Tate Special Interest Group (SIG), submit technical questions for our technicians to answer, and download useful utilities from our CompuServe service.

**Custom Support Utility Disk**

The Utility Disk, only available as part of the Professional package, contains programs and utilities that help you work with dBASE II, dBASE III PLUS, and Framework II more efficiently. Some of these programs were originally published in ***TechNotes***.

In addition, a "mini-expert system" is available on this disk to help you choose the best method for recovering a damaged dBASE database file. You can use several of the utilities to recover certain types of common data corruption and repair file damage.

### *Close-Up*, Remote Debugging Utility

Also included in the Professional Support Package is *Close-Up*, by Norton-Lambert, a remote communication and debugging utility. This utility allows an Ashton-Tate support technician to operate any Ashton-Tate product remotely on your computer, via modem. *Close-Up* provides personalized support and results in fast solutions to complex technical problems. This special service is billed at $75 per hour to your credit card.

You gain access to this remote utility with a file called CUSTOMER.COM, located on the Custom Support Utility Disk. As with the Utility Disk, *Close-Up* is only available if you subscribe to the Professional Support Package.

### Electronic Mailbox

If you have access to a modem, you can send us your data or program files quickly. Sometimes an intricate problem makes itself apparent when a technician can view it firsthand. Our electronic mailbox service provides that extra dimension that allows our technicians to respond quickly and accurately.

## Other Support Services

Authorized Ashton-Tate dealers (including Major Accounts), VARs, and Corporate Support and Services members who have valid access codes are entitled to direct access phone support via our *Express Support Queue*. For further information on these special service programs, call (213) 538-8880.

## Policy Information

The contract period for each package is one year. After a year, you may renew the package you originally signed up for, or switch to another package. If you allow either of your subscriptions to lapse (that is, if you do not renew before your one year of support expires), you will be charged a 25% re-subscription fee in addition to the cost of the package. For example, if your subscription to the Professional Support package expires on September 11, 1987, and you decide to renew it on October 9, you will pay $187.50 instead of $150.

When you purchase a support package, it is tied to a single product's serial number, not a single user. So if you purchased support for your Framework II package and you also want support on your dBASE III PLUS, you must purchase an additional package. Similarly, if you have five copies of dBASE III PLUS and you want support on all five, you need to buy five support packages. The rates for additional one-year support packages are:

| | |
|---|---|
| Extended Support: | $50 per year for 10 calls |
| | $80 per year for 20 calls |
| Professional Support: | $100 includes everything except ***TechNotes*** and CompuServe |

With the inception of the Custom Support Plan, we at Ashton-Tate are confident that we can meet your support needs now better than we ever could before.

# Custom Support Plan Q & A

The following Q & A offers you answers to some of the questions you may have about the Plan. If you need further information, call this number which has been specifically set up to answer any questions you may have about Custom Support: **(213)538-8880**. (Questions appear in *italics*.)

*Question*:
*Does either of the support plans include a toll-free 800 telephone number?*

Answer:
No. The costs associated with a toll-free, 800 number would significantly increase the cost of delivering support. We have instead chosen to increase our support staff. This will reduce the wait time for service so that any toll charges you incur will be kept to a minimum.

*Question*:
*I've called Ashton-Tate for support in the past and have had to wait 10 minutes or more on hold from long distance just to speak to someone about my problem. Why should I pay extra for this service now?*

Answer:
We have hired more than two dozen technical and administrative people since the beginning of 1986, thereby doubling our support staff. We feel confident that this increase in staff will, in turn, be able to provide you with better response time and better service in general.

We also feel that, by limiting our support services to our legitimate registered users, we can concentrate on the group of people who need and deserve our services most—our subscribers.

*Question*:
*What happens to my number of calls if I subscribe to one of the support packages, and I call with a question that turns out to be an anomaly in the software?*

Answer:
Our support technicians will determine whether or not your problem lies within the software. Any call made for the sole purpose of notifying us of or verifying a known software anomaly will not be counted as one of your allotted calls.

*Question*:
*What if I need to make several calls to solve a particular problem? Will those calls count?*

Answer:
Again, our support technicians will determine this. We maintain an on-line call history profile on each user. This allows us to make a reasonable determination on the proper completion of a particular problem. In general, follow-up calls on a single specific problem will not be counted as one of your allotted calls.

*Question*:
*I am a consultant and do not actually own a copy of dBASE or Framework. Do I have to be a registered user in order to call for support?*

Answer:
Yes. Only registered users will be allowed access to direct telephone support services. Other support services, such as *TechNotes* and our on-line information networks on The Source and CompuServe, are available to anyone.

*Question*:
*If I just bought your product and need assistance but have not yet had a chance to send in the registration card, will I get help if I need it?*

Answer:
Yes. Your authorized Ashton-Tate dealer is committed to assisting you in the set-up and installation of our software. If your dealer is unable to assist you, you can make one support call to the Software Support Center without having a registration card on file. Our special operators will ask you for some required information, pre-register you, and request that you return your registration card to us within 30 days from the date of your software purchase. Returning your registration card within 30 days of purchase will entitle you to 90 days of warranty support.

*Question*:
*I sent in my registration card a few months ago. Am I still eligible for phone support, and if so, for how much longer?*

Answer:
Yes. All users, registered or unregistered, will receive 90 days of free, unlimited, direct access phone support as of September 1, 1986.

*Question*:
*I lost my registration card. How can I get another one so that I can receive phone support?*

Answer:
Contact our Customer Support Representatives at (213) 538-8880. They will give you instructions on how to register your software.

*Question*:
*Can I make any substitutions in any of the services and products that are included in the Professional Support Package?*

Answer:
No. The items included as part of the Professional Support Package are fixed. If you already subscribe to ***TechNotes***, your existing subscription will be extended by an additional year. If you already subscribe to CompuServe, you will receive an additional subscription with this package.

*Question*:
*Can I examine a sample copy of* **TechNotes** before I subscribe, and if so, who should I contact?

Answer:
Yes, you can obtain a sample copy by calling (213)538-8880. (Only one sample copy per person, please.)

*Question*: *Do I need to be a registered user to subscribe to* **TechNotes**?

Answer:
No. Anyone may subscribe to ***TechNotes***. If you wish to subscribe independently of the Custom Support Plan packages, please call one of the following 800 numbers:

(800) 828-2514
(800) 331-4164 (in California)
(619) 487-3212 (in Canada)

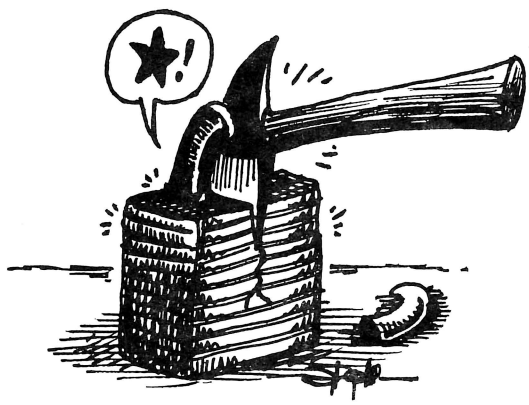
NOTE: THESE NUMBERS WILL DIAL DIRECTLY TO AN EXTERNAL MAGAZINE SUBSCRIPTION SERVICE FOR ASHTON-TATE PERIODICALS ONLY. CALLING THEM FOR ANY OTHER REASON WILL ONLY CAUSE UNNECESSARY DELAY.

* * * *

# No More Copy Protection!

On August 19, 1986, Ashton-Tate announced the removal of copy protection from its IBM PC, PC-compatible, and Macintosh computer products. New, non-copy-protected versions of dBASE III PLUS, dBASE III PLUS LAN Pack, and Framework II were shipped to Ashton-Tate authorized dealers. Registered owners of Ashton-Tate products can receive updates to these new products.

The new programs are dBASE III PLUS version 1.1, dBASE III PLUS LAN Pack version 1.1, and Framework II version 1.1. The MultiMate line of word processing products is not copy-protected. Copy protection will not be used in the company's line of products for the Macintosh personal computer. dBASE Mac, an advanced relational database management system for the Macintosh, will be available in the fourth quarter of this calendar year as a non-copy-protected release.



Registered users of dBASE III PLUS, dBASE III PLUS LAN PACK, and Framework II who purchased their software or upgraded from an earlier version after July 5, 1986, are entitled to free upgrades. Those who purchased or upgraded their software before July 5 will be charged $45 for the upgrade. Owners of dBASE III may upgrade to dBASE III PLUS version 1.1 and owners of Framework may upgrade to Framework II version 1.1 for $175 until December 1, 1986, and for $275 after that date. In addition, registered owners of dBASE II, Friday!, Bottom Line Strategist, and Financial Planner may upgrade to the non-copy-protected version 1.1 of dBASE III PLUS for $275. (Owners of these programs should be aware, however, that dBASE III PLUS version 1.1 is only supported on the IBM PC, PC XT, PC AT and 100% compatible computers.) Volume discounts are available, and unregistered users and others requiring assistance may call 1-800-2-ASHTON for information on receiving the upgrade.

# New Installation Procedures

Framework II version 1.1 and dBASE III PLUS version 1.1 require that you run the ID program on System Disk #1 to identify the owner of the software. You are prompted to enter your name, the company name, and the serial number printed on the System Disk #1 label. Every time you start dBASE III PLUS or Framework II, this information will be included in the first screen you see. The ID program can only be run once, and the information cannot be modified later. If you are not the owner of the software, contact the appropriate person for any special instructions before completing the ID screen. dBASE III PLUS and Framework II will not run unless you have entered the appropriate information in the ID screen first.

After running the ID program, you can use the DOS COPY command to install Framework II or dBASE III PLUS on your hard disk. For Framework II, the SETUP program is preferable to a DOS COPY because it allows you to customize your copy of Framework II to your equipment and applications and offers the option of installing on the hard disk. Similarly, the INSTALL program in dBASE III PLUS will copy the system files onto the designated hard disk. If you have an earlier version of dBASE III PLUS or Framework II in the hard disk directory into which you are installing, SETUP and INSTALL will delete the earlier version. Therefore, if you wish to save the previous version, uninstall it first using the version 1.0 disks from which the software was originally installed. SETUP or INSTALL will prompt you to execute the ID program if you have not already used it and entered the appropriate ID screen information.

As dBASE III PLUS version 1.1 and Framework II version 1.1 are not copy-protected, you can use the DOS BACKUP and RESTORE commands to make backups of all your data. The earlier copy-protected versions would not allow you to RESTORE the root directory of your hard disk without ruining the installed software.

If the hard disk becomes damaged or you need extra disk space and are not using dBASE III PLUS or Framework II, you can erase files with the DOS DEL command. Rather than using DEL, if you run Framework II SETUP and choose to uninstall, or run the dBASE III PLUS UNINSTAL program, only the program system files will be removed. Data files in the directory remain intact and can be removed later with DEL.

# Installing dBASE ADMINISTRATOR

Installing dBASE ADMINISTRATOR version 1.1 is similar to installing single-user dBASE III PLUS. The IDLAN program, which is on the Administrator #1 disk, should be run first; you will be prompted for the same information that the ID program requests. If you want to have both dBASE III PLUS version 1.1 and dBASE ADMINISTRATOR version 1.1 on the same hard disk, they must be installed into separate directories. dBASE ADMINISTRATOR can be installed by typing

```
INSTALL C: DBA
```

at the **A>** prompt with System Disk #1 in drive A. Unlike single-user, copying the files onto the hard disk will not allow access to the program; dBASE ADMINISTRATOR must be installed in order to run dBASE III PLUS. Though the files are not copy protected, access is controlled in the LAN environment by the installation process. During installation of the dBASE ADMINISTRATOR, a separate directory named DBNETCTL.300 is created on the hard disk. Files in this directory are used to log the current number of workstations that can run dBASE III PLUS on the LAN at one time.

Installation of dBASE ADMINISTRATOR enables one user to run the program. The ADDUSER program, which is located in the same directory as the dBASE ADMINISTRATOR after installation, allows you to add additional users. Each dBASE Access disk (from the dBASE III PLUS LAN Pack, which is purchased separately) allows you to add five more users. You may also use a System Disk #1 or an Access disk from previous versions of dBASE III PLUS to add users. Each System Disk #1 or Access disk allows you to add one user. ADDUSER will optionally reduce the number of users and list the current number of users installed on the network.

With this new multi-user installation process, the number of users is tied to the LAN, not to each node on the network. Copy protection and access to the network have been decoupled and each user does not need to have a key disk or installed Access to log onto dBASE III PLUS. This also permits diskless workstations.

The removal of copy protection and the Custom Support Plan are both aimed at providing our users with more convenient ways to use our software.

# dBASE III PLUS Change Summary—Part Four

*by Joseph Perret*

This is the fourth and last part of the four-part change summary listing anomalies reported in ***TechNotes*** for dBASE III version 1.1 and the Developer's Release fixed in dBASE III PLUS. This part lists anomalies reported from the April through the June 1985 issues of ***TechNotes***. As was the case in each of the last three parts, each entry is followed by an explanation of the change made to the command or function. Some anomalies have been repaired outright and now work as documented. Others, occurring when an internal limit was breached or incorrect syntax employed, now produce appropriate error messages.

## ?/??

Formerly, any attempt to display a memo field from a work area other than the currently SELECTed work area produced the error message "Unrecognized phrase/keyword in command."

In dBASE III PLUS you can LIST or DISPLAY a memo field from a work area other than the SELECTed work area but you still cannot access a memo field with ? or ?? without error.

Reported in the May 1985 issue of ***TechNotes***, page D3-14.

## @...GET...READ

1. Formerly, when you entered numbers to the right of the decimal place of numeric fields, typing the decimal point would sometimes cause the cursor to jump back to the beginning of the field. The problem occurred using @...GET...READ, APPEND, BROWSE, CHANGE, or EDIT on a database file with one or more index files in USE.

In the Developer's Release and dBASE III PLUS, when you type the decimal point, you can properly enter the decimal digits of your number.

Reported in the April 1985 issue of ***TechNotes***, page D3-28.

2. Formerly, if you attempted to GET a memory variable with the PICTURE clause "999.99" and you did not initialize the memory variable with the number of decimal places specified in the PICTURE clause, the GET display did not show the decimal digits for input. If you entered a decimal point, you would exit the GET.

   In the Developer's Release and dBASE III PLUS, READ allows the PICTURE clause to determine the number of decimal places in a GET, regardless of the number of decimal places the memory variable is initialized to.

   Reported in the May 1985 issue of ***TechNotes***, page D3-1.

# @...SAY...GET

Formerly, if you displayed text to the screen with @...SAY, and then did an @...SAY...GET to the same line without erasing it first, characters from the previous text would appear between the SAY and the GET. For example,

```
STORE "123" TO mem
@ 10,0 SAY "XXXXXXXXXXXX"
@ 10,0 SAY "A" GET mem
```

displayed

```
AX123XXXXXXX
 ↑
 └──────────── This should not be displayed.
```

In dBASE III PLUS, an @...SAY...GET clears the screen between the SAY and GET as expected.

Reported in the April 1985 issue of ***TechNotes***, page D3-29.

# APPEND

When you use APPEND, you can move back into existing records and edit them. Previously, if an index file was in USE and you edited key fields in existing records, the index file was not updated with the new key field values. You had REINDEX to update the index file.

In the Developer's Release and dBASE III PLUS, changes to key fields made in this way are reflected in the index and no REINDEX is necessary after you leave APPEND.

Reported in the April 1985 issue of ***TechNotes***, page D3-29.

# APPEND FROM

Formerly, if you APPENDed FROM a DELIMITED file that contained a space after a numeric field and before the comma, a zero would appear in the field following the numeric field. The blank space was apparently read as a separate field with a value of zero.

In the Developer's Release and dBASE III PLUS, the inclusion of a blank space between the numbers in a numeric field and the field separator does not interfere with the ability of APPEND FROM DELIMITED to correctly APPEND data.

Reported in the June 1985 issue of ***TechNotes***, page D3-20.

# APPEND FROM SDF

APPEND FROM SDF expects records in the source text file to be terminated with the ordered pair, carriage return/linefeed (0DH 0AH). Formerly, if the order was reversed (0AH 0DH), dBASE III ignored the character following the carriage return and the first character of every record after the first record was missing in the target database file.

In the Developer's Release and dBASE III PLUS, a reversal of the positions of carriage return and linefeed does not affect APPEND FROM SDF.

Reported in the May 1985 issue of ***TechNotes***, page D3-2.

# ASSIST

1. Formerly, if you chose an option in ASSIST that allows a FOR/WHILE clause (AVERAGE, DISPLAY, REPORT, etc.) on a database file with exactly 27 fields, you could not access the last several fields as part of the FOR/WHILE clause with **Dnarrow**. Everything appeared fine until the last group of fields was displayed on the screen, at which point pressing **Dnarrow** took you to the next menu screen.

   In the Developer's Release and dBASE III PLUS, all fields may be accessed through ASSIST, regardless of the number of fields in the database file.

   Reported in the April 1985 issue of ***TechNotes***, page D3-29.

2. Formerly, if SET INTENSITY was OFF, you could not see the available options in ASSIST. This happened because ASSIST relied on the options being in reverse video and yet neglected to change the status of the intensity parameter.

   In the Developer's Release and dBASE III PLUS, the status of SET INTENSITY does not affect operations in ASSIST.

   Reported in the April 1985 issue of ***TechNotes***, page D3-30.

# AVERAGE/SUM

1. Formerly, SUM and AVERAGE sent an extra linefeed to both the screen and printer upon command termination.

   In the Developer's Release and dBASE III PLUS, SUM and AVERAGE send only one linefeed upon termination.

   Reported in the April 1985 issue of ***TechNotes***, page D3-35.

2. Formerly, attempting to SUM TO memory variables declared PUBLIC usually gave erroneous results. Initializing the variables before the SUM command would sometimes, but not always, correct the error. In addition, there was a practical limit to the number of expressions that could be used in one SUM command.

   In order to use the SUM command reliably, the following rules had to be followed.

   a. SUM TO PRIVATE variables and then STORE the results to PUBLIC variables.

   b. Limit the number of expressions in the SUM command to three or less.

In the Developer's Release and dBASE III PLUS, PUBLIC does not affect the results of SUM or AVERAGE. The number of expressions possible is limited only by the size of the command line.

Reported in the April 1985 issue of ***TechNotes***, page D3-36.

## CLOSE PROCEDURE

Formerly, if you included CLOSE PROCEDURE or SET PROCEDURE TO [ < filename > ] in a procedure file, dBASE III would stop running and you would have to warm boot your computer.

The Developer's Release and dBASE III PLUS return the error message "File is already open" when this occurs.

Reported in the April 1985 issue of ***TechNotes***, page D3-30.

## CONFIG.DB

Formerly, the CONFIG.DB file would not accept WP or TEDIT parameters more than eight characters in length. Any more than eight would be truncated. Attempting to use MODIFY COMMAND (or to edit a memo field) briefly displayed the operating system message "Bad command or file name" and dropped you to the dBASE dot prompt or the EDIT screen.

In the Developer's Release and dBASE III PLUS, the WP and TEDIT parameters in CONFIG.DB can be any length, as long as they remain on one line of text.

Reported in the April 1985 issue of ***TechNotes***, page D3-30.

## COPY TO DELIMITED

Formerly, if a database file structure contained a large number of character or numeric fields with a length of one, COPY TO DELIMITED would produce unreliable results.

In the Developer's Release and dBASE III PLUS, COPY TO < filename > DELIMITED properly copies one-character fields up to the 128-field limit.

Reported in the June 1985 issue of ***TechNotes***, page D3-20.

## COPY TO SDF

Formerly, if you COPYed a database file with a memo field using the SDF option, data in the fields following the memo field either would not be copied or would be copied incorrectly. If, however, you specified the fields following the memo field in the FIELDS clause, the problem did not occur.

In the Developer's Release and dBASE III PLUS, you can COPY TO SDF fields following a memo field.

Reported in the April 1985 issue of ***TechNotes***, page D3-30.

## CREATE

Formerly, if you CREATEd a database file, you had to USE it to establish the filename as the default alias. Otherwise, the database file had no alias assigned to it and referencing the alias returned the error message "Alias not found."

In the Developer's Release and dBASE III PLUS, the filename becomes the default alias as soon as the database file is CREATEd, making it unnecessary to USE the database file again to assign the alias.

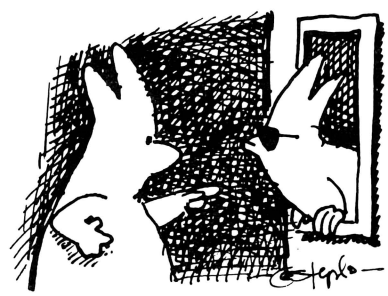
Reported in the April 1985 issue of ***TechNotes***, page D3-30.

## Ctrl-C While Printing

Formerly, if you pressed **Ctrl-C** while printing, you were returned to the operating system. LIST TO PRINT, DISPLAY TO PRINT, and operations performed with SET PRINT ON and REPORT FORM TO PRINT were particularly affected.

In the Developer's Release and dBASE III PLUS, pressing **Ctrl-C** during a print operation does not affect the printout or return you to the operating system.

Reported in the June 1985 issue of ***TechNotes***, page D3-20.



## DO WHILE

1. Formerly, if you continued a DO WHILE statement to a second line with a semicolon, dBASE III tried to execute the second line the second time through the loop.

   In the Developer's Release and dBASE III PLUS, a DO WHILE command line wrapped with a semicolon executes properly each time through the loop.

   Reported in the April 1985 issue of ***TechNotes***, page D3-31.

2. Formerly, a RETURN inside a DO WHILE...ENDDO construction did not close the DO WHILE <condition> in the program. Therefore, the condition continued to be evaluated. For example,

```
*---A.PRG
expA = "1"
DO WHILE expA = "1"
    ? "Hi!"
    DO B ------------------------> *---B.PRG
ENDO                               expB = "2"
* EOP A.PRG                        DO WHILE expB = "2"
                                       ? "How are you?"
                                       expA = "X"
                                       RETURN
                                   ENDDO
                                   * EOP B.PRG
```

   Executing A.PRG caused an infinite loop. dBASE III continued to test the condition in B.PRG. To work around this, B.PRG had to be written as follows:

```
*---B.PRG (revised).
expB = "2"
DO WHILE expB = "2"
    ? "How are you?"
    expA = "X"
    EXIT <-------------------- Notice the EXIT here,
ENDDO
RETURN <------------------- and the RETURN here.
* EOP B.PRG
```

In the Developer's Release and dBASE III PLUS, a RETURN from within a DO WHILE loop properly closes that loop and the condition is no longer evaluated.

Reported in the April 1985 issue of ***TechNotes***, page D3-31.

# FILE()

Formerly, the FILE() function searched only the current directory and SET PATH TO did not affect the file search. If other directories were to be searched, the path had to be supplied as a part of the function argument.

In the Developer's Release and dBASE III PLUS, if the file is not found in the current directory, the FILE() function searches the path set by SET PATH TO for its existence.

Reported in the April 1985 issue of ***TechNotes***, page D3-3.

# FIND

Formerly, FIND &memvar would return "No find" if there were trailing spaces following the macro memory variable. This occurred whether the FIND was in a command file or issued at the dot prompt. Trailing blanks were concatenated to the macro memory variable when the FIND command was used. With the additional trailing blanks, the new character string might not be found.

In the Developer's Release and dBASE III PLUS, FIND &memvar acts identically to SEEK memvar. Trailing blanks are not included in the character string returned by the macro function.

Reported in the April 1985 issue of ***TechNotes***, page D3-32.

# INDEX

1. Formerly, if you attempted to create an index file with a key expression consisting of the concatenation of several STR() functions, INDEX ON would create an index file in record number order instead of key order. Specifically, this would occur if the length of the key expression exceeded 80 characters and consisted solely of concatenations of STR() functions.

   In the Developer's Release and dBASE III PLUS, it is possible to INDEX ON any legal expression regardless of the number of STR() functions used.

   Reported in the May 1985 issue of ***TechNotes***, page D3-12.

2. Formerly, the maximum length of a key expression and the key itself was 100 characters. If the key or the key expression exceeded the 100-character limitation, dBASE III would normally return the error message "Index is too big (100 char maximum)."

   However, this error was not trapped if the key expression was formed of the concatenation of STR() functions. If you attempted this no error message was returned, but one of two errors would occur. Either the index created was still in record number order, as explained above, or dBASE III would return random error messages as it attempted to create the index file.

   In the Developer's Release and dBASE III PLUS, the limitation on the length of an index key expression is 220 characters. Any key expression that exceeds this limit returns "Index expression too big (220 char maximum)."

   Reported in the May 1985 issue of ***TechNotes***, page D3-12.

# LABEL

Formerly, LABEL FORMs of 13 or more lines could freeze the computer. Additionally, shorter forms would not print the last few lines when these lines contained a great deal of text. The problem stemmed from the maximum complexity of a label, which is mainly controlled by its size.

In the Developer's Release and dBASE III PLUS, LABEL FORM successfully outputs labels to its limit of 16 lines per label.

Reported in the April 1985 issue of ***TechNotes***, page D3-33.

# MODIFY REPORT

Formerly, executing a REPORT FORM with a width and a left margin whose difference was exactly 256 displayed garbage on the screen or printer and froze the computer requiring a warm boot. Increasing the width or decreasing the left margin so that the difference was greater than 256 corrected this problem, but the report title displayed on several lines instead of one. A difference of 255 or less displayed the report correctly.

In the Developer's Release and dBASE III PLUS, a REPORT FORM with a width of exactly 256 prints correctly, with the exception that the report title is not printed. If the report width is increased, the title is output in the number of columns based on the formula

    Report Width - 256

Reported in the April 1985 issue of ***TechNotes***, page D3-33.

# RECNO()

Formerly, RECNO() returned the number of the last record accessed in the current work area prior to closing the database file. Even if the database file was closed, RECNO() retained this last value.

In dBASE III PLUS, RECNO() is reset to zero as soon as the database file is closed.

Reported in the May 1985 issue of ***TechNotes***, page D3-14.

# RELEASE ALL

Formerly, there were instances where RELEASE ALL did not work. After executing a DO WHILE loop < n > times (the reported instances have occurred at three iterations), the message "Out of memory variable memory" appeared, although the memory limits were not exceeded. When RELEASE ALL was replaced with CLEAR MEMORY, the problem disappeared.

This error was originally reported in version 1.0 of dBASE III. It has not been reported in dBASE III version 1.1, the Developer's Release, or dBASE III PLUS.

Reported in the May 1985 issue of ***TechNotes***, page D3-15.

# REPORT FORM

MODIFY REPORT allows the number of decimal places to be changed from the default. If you do this and run the report, everything proceeds as expected. Formerly, however, if you MODIFYed a REPORT a second time, the number of decimal places reverted to the default when the cursor reached the "# decimal places" field.

In the Developer's Release and dBASE III PLUS, the number of decimal places in a column definition remains at its given value until explicitly altered.

Reported in the April 1985 issue of ***TechNotes***, page D3-34.

# SET CARRY

Formerly, if you added records to a database file with SET CARRY ON and the last record in the database file was marked for deletion, the new records were also marked for deletion. This condition was true whether you added records with APPEND or INSERT.

In dBASE III PLUS, the deletion marker is no longer carried forward during an APPEND or INSERT, regardless of the status of SET CARRY.

Reported in the April 1985 issue of ***TechNotes***, page D3-34.

# SET DEBUG ON

Formerly, SET DEBUG ON overrode SET CONSOLE OFF and output was echoed to the screen.

In the Developer's Release and dBASE III PLUS, SET DEBUG ON does not output to the screen, regardless of SET CONSOLE.

Reported in the June 1985 issue of ***TechNotes***, page D3-3.

# SET DELIMITERS

Formerly, if you SET DELIMITERS ON, then CREATEd or MODIFYed a database file structure, pressing **Esc** without making any changes SET DELIMITERS OFF.

In the Developer's Release and dBASE III PLUS, CREATE and MODIFY STRUCTURE do not affect the status of SET DELIMITERS.

Reported in the April 1985 issue of ***TechNotes***, page D3-34.

# SORT

1. Formerly, you could not SORT TO a directory other than the current one. Attempting to do this did not create the designated target database file or produce an error message. Instead, a temporary database file with the extension name .W44 would be created in the current directory of the default drive.

   In the Developer's Release and dBASE III PLUS, SORT successfully creates a database file in a directory other than the current directory.

   Reported in the April 1985 issue of ***TechNotes***, page D3-34.

2. Formerly, SORT using a scope other than the default, ALL, would create a target database file with an incorrect number of records.

   In dBASE III PLUS, a scope may be included in a SORT command and the target database file is correctly constructed.

   Reported in the April 1985 issue of ***TechNotes***, page D3-35.

3. Formerly, the effective limit of SORT was in the 32,000 record range.

   The work-around for sorting database files in excess of 32,000 records was to INDEX and then COPY, although this would not produce descending order.

   In the Developer's Release and dBASE III PLUS, you can SORT files larger than 32,000 records without error.

   Reported in the April 1985 issue of ***TechNotes***, page D3-35.

4. Formerly, if you attempted to SORT with the condition, SUBSTR( <field name>, <n>, <n>) = <value>, returned the error message "00% Sorted *** STACK OVERFLOW ***" and dropped you to the operating system.

   In the Developer's Release and dBASE III PLUS, the SUBSTR() function may be used in a conditional clause of the SORT command without error.

   Reported in the June 1985 issue of ***TechNotes***, page D3-21.

# STORE

Formerly, if you attempted to STORE a number with more than 40 digits to a memory variable, the computer might freeze requiring a warm boot. STOREing a number with more than 60 digits would freeze the computer requiring a cold boot.

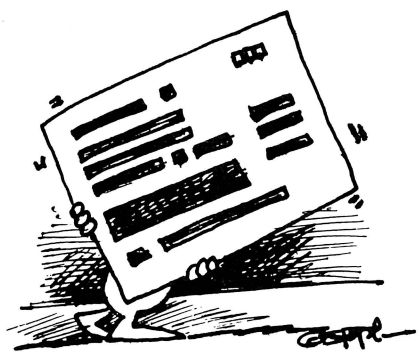
In the Developer's Release and dBASE III PLUS, STOREing a very large number to a memory variable STOREs a numeric overflow. The Developer's Release STOREs .9999999999E99 and dBASE III PLUS STOREs asterisks.

Reported in the April 1985 issue of ***TechNotes***, page D3-35.

# Simulating the dBASE Picture Clause for Data Input

*by Karen Robinson and Kent Irwin*

One of the primary reasons for writing a custom data entry program is to control the way in which data is entered. This article discusses a user-defined function, Picture.FW2 that allows you to enhance your own data entry program for displaying and validating data. By using Picture, you can ensure that information is always entered in the same format and does not contain inappropriate values. In addition, you can format the display of character data, accept numeric input only, or accept a mix of numbers and letters. For example, you can format a phone number as (  )  -  . As you enter the phone number, the parentheses and dash characters are automatically inserted. For numeric data you can perform a range check, limiting the size of the number entered by specifying an upper and lower limit of acceptable values.



## Parameters

Picture accepts one parameter for character data and three parameters for numeric data. Each parameter is passed as a character string. To execute Picture, use the following generalized syntax:

@Picture( < format > , < lower limit > , < upper limit > )
used with numeric data only

The format parameter defines the maximum length and data type (character or numeric) of the spreadsheet or database cell input. The number of characters in the format determines the number of input digits that will be accepted. The data type of accepted input is determined by the first character of the format parameter. "9" limits you to numeric data and "X" to character data. The format characters "9" and "X" also serve to mask input for number or character digits. Any other character you use as part of the format string displays as a literal value and is considered part of the data. Examples of literals are the parentheses and dash characters used in a phone number. The following illustrates the syntax of Picture for entering a phone number as a character string with embedded formatting characters:

@Picture("(999)999-9999")

The parentheses and dash characters are included as part of the data. Once a phone number is entered, the cell looks like this:

(213)329-0086

The second and third parameters of Picture allow you to define a continuous range for numeric data by specifying lower and upper boundaries. The lower limit is a value that defines the smallest numeric value to accept for input. The upper limit is a value that defines the largest numeric value to accept for input. For example, the following shows Picture syntax for inputting a three-digit numeric with a minimum input value of 10 and a maximum of 20:

@Picture("999", 10, 20)

# Setup

Picture is called by your data entry program. It assumes that the cursor is located on the blank cell you want enter data into. If the data input is to be numeric, it also assumes that you are in **EDIT FORMULA (F2)**. At this point you execute Picture to accept and format the data. Once the data for the current cell is entered, control returns to your program. You then navigate to the next cell for data input and call Picture again. This process repeats for each entry.

To set up Picture, follow these steps:

1. Create an empty word frame and label it Picture.
2. Then, enter the program code (Picture.FW2) in the formula area of the frame.
3. In your data entry program, call the Picture function using the format describe above.

The following are two examples of how to use Picture. The first sample program Database.FW2, demonstrates how to Picture would be used to enter data into a database; the second Sheet.FW2 demonstrates it in a spreadsheet application. To use these sample programs, follow the steps below:

1. Create a two-field database and label it Db.
2. Label the first field Name and the second Amount.
3. Then create a two-column spreadsheet and label it Ss.
4. Label the first column Name and the second Amount.
5. Create two word frames. Label one Database and the other Sheet.
6. Enter the programs, listed below, into the formula area of the appropriate frames.
7. Before executing either program, be sure the database or spreadsheet is in Frames View.

To enter data into the database, Db, highlight the frame labeled Database, and press **RECALC (F5)**. To enter data into the spreadsheet, Ss, highlight the frame labeled Spreadsheet, and press **RECALC (F5)**.

## Picture With a Database

```
; Database.FW2
;
;   by Kent Irwin
;   October 1, 1986
;
; Highlight the database.
@setselection("Db"),
@isabend(#FALSE),
; Move the cursor to the first empty record.
@if(@get(Db.Name) = #N/A!,
     @performkeys("{Dnlevel}{Ctrl-Home}{Dnarrow} "),
     @performkeys("{Dnlevel}{Ctrl-Home}{Ctrl-Dnarrow}{Dnarrow} ")
),
; If end-of-file, create a blank record.
@if(@isabend,
    @performkeys("{Esc}{Ctrl-C}R1{Return} ")
),
@eraseprompt,
@prompt("Please enter name",32),
; Execute Picture routine and limit field size to 30 characters.
@Picture("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"),
; Move to next field.
@performkeys("{Rightarrow}{F2}"),
@eraseprompt,
@prompt("Please enter Amount (0 - 500)",25),
@Picture("999", 0, 500),
@performkeys("{Ctrl-Uplevel}")
```

## Picture With a Spreadsheet

```
; Sheet.FW2
;
;   by Kent Irwin
;   October 1, 1986
;
```

```
@setselection("Ss.A2"),
@eraseprompt,
@prompt("Please enter Name",32),
@Picture("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"),
@performkeys("{Rightarrow}{Del}{F2}"),
@eraseprompt,
@prompt("Please enter Amount (0 - 500)",25),
@Picture("999", 0, 500),
@performkeys("{Ctrl-Uplevel}"),
@prompt("ok")
```

---

---

```
; Picture.FW2
;
;    by Kent Irwin
;    October 1, 1986
;
; Pass the following parameters to Picture:
;
; @item1 ::= picture string
;            9 - accepts digits only
;            x - accepts any character
;            all other picture characters are forced literals
; @item2 ::= lo range <optional>
; @item3 ::= hi range <optional>
;
@local(index,temp,loop,num),
@if(@mid(@item1,1,1) = "9",
    @if(@itemcount <> 3,
        @list(
            @eraseprompt,
            @prompt("PICTURE: Wrong number of arguments!!",22),
            @return(0)
          )
        ),
        @if(@itemcount <> 1,
           @list(
               @eraseprompt,
               @prompt("PICTURE: Wrong number of arguments!!",22),
               @return(0)
             )
        )
),
index := 1,
loop := #TRUE,
@while(loop = #TRUE,
   @while(@and(index <= @len(@item1),@key <> {Return}),
        temp := @mid(@item1,index,1),
        @if(@or(temp = "x",temp = "9"),
            @if(temp = "9",
                @list(
                    @while(@and(
                                @iserr(@value(@keyname(@nextkey))),
                                @keyname(@key) <> ".",
                                @keyname(@key) <> "-"
                           ),
```

```
                        @beep(292,10)
                    ),
                    @performkeys(@keyname(@key))
                ),
                @performkeys(@keyname(@nextkey))
            ),
            @performkeys(temp)
        ),
        index := index + 1
    ),
    @if(@key <> {Return},
        @while(@nextkey <> {Return},
            @beep,
            @eraseprompt,
            @prompt("Please press Return",30)
        )
    ),
@performkeys("{Return}"),
@if(@mid(@item1,1,1) = "9",
    @if(@iserr(@textselection),
        @list(loop := #FALSE, @performkeys("{Return}")
        ),
        @if(@or(@value(@textselection) < @item2,
                @value(@textselection) > @item3
            ),
            @list(
                @performkeys("{Del}"),
                index := 1,
                @beep(392,10),
                @eraseprompt,
                @prompt("Value too high or too low",27)
            ),
            @list(loop := #FALSE, @performkeys("{Return}"))
        )
    ),
    loop := #FALSE
  )
),
@eraseprompt
```

# Conclusion

The Picture function presented here demonstrates how you can use FRED to preserve the integrity of your data by formatting and validating information as it is entered. If you want to add enhancements to Picture, study it and change it as you see fit.

# Optimizing Network Techniques

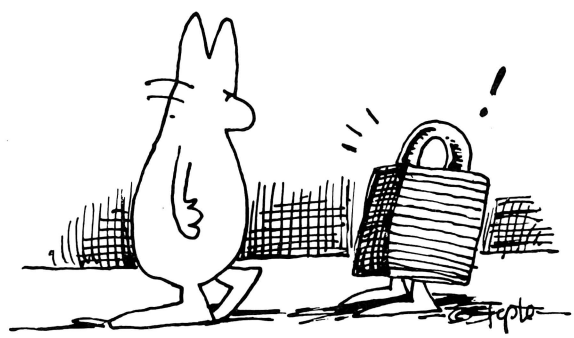
*by J. Mike Hedblom*

## Avoiding File Locking with Network Operations

Some commands require the EXCLUSIVE USE of a database file to execute properly. You have probably found that it is a significant problem if several people are attempting accessing the same database file at the same time in order to use a command that requires a file lock or EXCLUSIVE USE of the database file. This article presents a number of ideas and strategies that can help you manage your database file access environment.

One of the most frequently used commands that requires EXCLUSIVE USE of a database file is APPEND. In a system that requires APPENDing every day, one way to get around the file access problem is to start the day by APPENDing as many blank records as are likely to be needed for that day's activities. If, for example, you average 20 new records a day, APPEND 20 blank records to the database file in the morning before the access becomes too heavy for EXCLUSIVE USE. At the end of the day, you can DELETE all the blank records that haven't been used or leave them for the next day. This can be done quite simply with the following procedure:

```
* ---Add 20 blank records to current database file.
addrecs = RECCOUNT() + 20
DO WHILE RECCOUNT() < addrecs
    APPEND BLANK
ENDDO
RETURN
```



# APPENDing to a Database File in a Network Environment

This procedure works well if you have allowed for enough blank records at the start of the day. You may, however, have a busy day and discover that you need to add more records than you initially allocated When this happens you will have to use the locking functions to gain access to the database file when it becomes available. The following program, Startday.PRG, is an example of a daily start-up routine to add blank records that can additionally deal with a database file during periods of high usage.

```
* Program ...: Startday.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: APPENDs 20 records to a shared database file.
*
SET TALK OFF
SET EXCLUSIVE ON
PUBLIC tries
CLEAR
ON ERROR DO Err
* ---"tries" will be used by Err if an error occurs.
tries = 0
* ---If an error occurs while USEing Tax.DBF,
* ---Err.PRG RETRYs to USE it.
USE Tax
? "APPENDing records..."
cnt = 0
DO WHILE cnt < 20
   APPEND BLANK
   cnt = cnt + 1
ENDDO
? "Done APPENDing."
USE
SET EXCLUSIVE OFF
ON ERROR
SET TALK ON
RETURN
* EOP Startday.PRG
```

```
* Program ...: Err.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: Err.Prg is an error trap for Startday.PRG.
*              It is currently only intended to trap the error,
*              File in use by another." For other errors it
*              displays the error message.
*
DO CASE
CASE ERROR() = 108 &&Error 108 is "File in use by another."
     * ---If the file is in use, RETRY to USE the file
     * ---five times. After the fifth attempt, abort
     * ---the request and try again later.
     DO WHILE tries < 5 .AND. ERROR() = 108
        tries = tries + 1
```

```
        @ 10,20 SAY "Tries: " + STR(tries,2,0)
        RETRY
    ENDDO
    IF tries = 5
        ? CHR(7)
        CLEAR
        @ 10,20 SAY message()
        @ 12,20 SAY "Try again later."
        CANCEL
    ENDIF
OTHERWISE
    * ---Some other error has occurred. Display that error
    * ---and wait.
    ? ERROR()
    ?? MESSAGE()
    WAIT
ENDCASE
RETURN
* EOP Err.PRG
```

## Orderly Access to Database Files for Network Systems

One method of USEing a database file EXCLUSIVEly while still being fair to all users of a system is to manage file requests with a program. Since memory variables cannot be shared, posting a request must be done through a shared-process database file. The structure of this global database file should contain fields for the filename, type of request, and some kind of code to identify the requestor. The database file will only be accessed by a task management program.

The program must be able to flag requests for a database file and identify whether the request requires EXCLUSIVE USE. The identifying code is determined with the TIME() function, since only one user can post a request at any given time. The following is the structure of the global control database file:

**Structure for database: Opt_file.DBF**

| Field | Field name | Type | Width |
|---|---|---|---|
| 1 | Filename | Character | 12 |
| 2 | Ex_request | Logical | 1 |
| 3 | File_id | Character | 8 |

All changes to it are maintained by a file management program to insure the integrity of the process. This means that all shared database files must be opened by the file management program in order to maintain the credibility of the system. When the dBASE Administrator opens a database file, it must check to see if it is already in USE. If the database file is in USE, then it must determine if the database file is being USEd under conditions that require EXCLUSIVE USE. An EXCLUSIVE operation means that a process is being performed that affects the entire database file and so all other users must be locked out. To a user trying to gain access to a database file under these conditions, the dBASE Administrator returns the error message "File is in use by another." If more than one user is sharing a database file and one of them tries to perform an EXCLUSIVE operation, the error message "Exclusive use on database is required" results and the operation is canceled. To keep the user from encountering these errors, the file management system must be able to determine if a request requires EXCLUSIVE database file USE. If the request is EXCLUSIVE, all other users of the requested database file must be put on hold as they finish their jobs. This allows an EXCLUSIVE request to gain access to the database file while the other users wait. When the exclusive request is finished, the other users then regain access to the database file.

The file management system is intended for systems that have many users sharing the same database files at the same time. If two people are using a database file, they can easily agree to terminate an operation so that an exclusive operation may be performed. This would be inconvenient in a large facility where there is continuous file activity. Therefore, all file accesses must be maintained by program control. Operations like APPEND, EDIT, and BROWSE are not allowed within this system since they cannot be controlled effectively. APPENDing should be done either to a temporary database file and later APPENDed to the master database file or by REPLACEing into existing blank records as described above. Record EDIT procedures should be done by GOing to the requested record and using CHANGE NEXT 1.

The following is an example of a database file access management program. It gives you the basic logic and coding necessary to initiate a file request system. It does not perform any record locking since there are a number of examples in the *Programming with dBASE III PLUS* manual.

# Program Listing

```
* Program ...: Optim.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: Main program for optimizing network
*              techniques.
*
SET TALK OFF
PUBLIC opt_choice, opt_file_n, opt_file_r, opt_file_i
PUBLIC wait_state, done
opt_file_n = " "
done = .F.
DO WHILE .NOT. done
   opt_choice = " "
   * ---Display the main menu and get user's choice.
   DO Opt_main
   IF opt_choice = "Q"
      done = .T.
      LOOP
   ENDIF
```

```
   * ---Is it the user's choice to work with a new file?
   IF opt_choice = "N"
      DO Opt_file
      LOOP
   ENDIF
   opt_file_r = .F.
   * ---Does the request require EXCLUSIVE USE?
   IF AT(opt_choice,"ARG") <> 0
      opt_file_r = .T.
   ENDIF
   * ---Reset the wait state of the request procedure.
   wait_state = 0
   * ---Request USE of the database file.
   DO Opt_req
   CLEAR
   * ---If the wait state returned by the Opt_req module
   * ---is 0, then the request has been completed and
   * ---work may now be done on the database file.
   IF wait_state = 0
      DO CASE
      CASE opt_choice = "A"
         ? "Append records."
      CASE opt_choice = "E"
         * ---Edits must be done with the CHANGE NEXT 1
         * ---command and not EDIT! EDIT doesn't stay
         * ---at one record.
         ? "Edit records."
      CASE opt_choice = "D"
         ? "Deletions."
      CASE opt_choice = "R"
         ? "Report forms menu."
      CASE opt_choice = "G"
         ? "Other Global file work."
      CASE opt_choice = "L"
         ? "Other Local file work."
      ENDCASE
   ENDIF
   * ---The file request is complete, now release it.
   DO Close_req
   * ---Release the error trap.
   ON ERROR
   CLOSE PROCEDURE
ENDDO
SET TALK ON
CLEAR ALL
RETURN
* EOP Optim.PRG
```

---

---

```
* Program ...: Opt_main.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: This is the main menu module for Optim.PRG.
*
main_choice = .F.
DO WHILE .NOT. main_choice
   CLEAR
   * ---If there is a file already in use, display
   * ---the current file name.
   IF LEN(TRIM(opt_file_n)) > 2
```

```
      @ 01,05 SAY "Current file is " + opt_file_n
   ENDIF
   opt_choice = " "
   @ 04,23 SAY "OPTIMIZE MAIN MENU"
   @ 06,20 SAY "Append . . . . . A"
   @ 07,20 SAY "Edit . . . . . E"
   @ 08,20 SAY "Deletions . . . . . .D"
   @ 09,20 SAY "Reports . . . . . .R"
   @ 10,20 SAY "Global file update . . G"
   @ 11,20 SAY "Local file request . . L"
   @ 14,20 SAY "New file . . . . . N"
   @ 16,20 SAY "Quit Optimize . . . . .Q"
   @ 19,22 SAY "Choice --> ";
           GET opt_choice;
           PICTURE "!"
   READ
   * ---Validate that the choice made is a valid one.
   IF AT(opt_choice,"AEDRGLNQ") = 0
      @ 21,5 SAY "*** Bad choice *** --> " + opt_choice
          LOOP
   ENDIF
   main_choice = .T.
ENDDO
RETURN
* EOP Opt_main.PRG
```

---

---

```
* Program ...: Opt_file.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: Procedure to determine file to use in Optim.PRG
*
fdone = .F.
DO WHILE .NOT. fdone
   ofilename = SPACE(8)
   CLEAR
   * ---Query for the name of the database file to work on.
   @ 10,15 SAY "Enter filename <filename> "
   @ 10,48 GET ofilename PICTURE "!!!!!!!!"
   READ
   opt_file_n = TRIM(ofilename) + ".DBF"
   * ---If the file does not exist, query for    retry.
   IF .NOT. FILE("&opt_file_n")
      qd = "N"
      @ 20, 5 SAY "File " + opt_file_n + " does not exist"
      @ 21, 5 SAY "Do you want a directory?"
      @ 23,15 SAY "Yes or No? ";
              GET qd;
              PICTURE "Y"
      READ
      * --- Does the user want a directory listing?
      IF qd = "Y"
         DIR
         WAIT
      ENDIF
      LOOP
   ENDIF
   fdone = .T.
ENDDO
RETURN
```

```
* EOP Opt_file.PRG
```

---

---

```
* Program ...: Opt_proc.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: Procedures for Optim.PRG
PROCEDURE Open_req
*
* Note(s) ...: Opens a file request by adding a record to
*              the list of database files in USE.
*
APPEND BLANK
REPLACE filename WITH opt_file_n
REPLACE ex_request WITH opt_file_r
REPLACE file_id WITH opt_file_i
* ---Release Opt_file.DBF
USE
RETURN
* EOP open_req
```

---

---

```
PROCEDURE Close_req
*
* Note(s) ...: Closes a file request by deleting it from the
*              list of database files in USE.
*
USE Opt_file EXCLUSIVE
GO TOP
LOCATE FOR filename = opt_file_n .AND. ;
          file_id = opt_file_i
DELETE
PACK
USE
RETURN
* EOP close_req
```

---

---

```
PROCEDURE Opt_wait
*
* Note(s) ...: Opt_wait releases the list of files that are in
*              use and then puts the user into a time-out loop.
*              At the end of the time-out, the wait state is
*              incremented to indicate that a wait has occurred.
*
USE
cnt = 0
DO WHILE cnt < 50
      cnt = cnt + 1
ENDDO
wait_state = wait_state + 1
RETURN
* EOP opt_wait
```

---

```
PROCEDURE Opt_err
*
* Note(s) ...: Opt_error is an error trap currently intended to
*              catch only those file that are currently in use
*              by another. In this program example, only the
*              file Opt_file.DBF should be in use since this
*              program will avoid all other collision. The
*              only way that this file could be in use is if
*              another request is being made. Therefore, a
*              RETRY is all that is needed to trap this
*              error. Should some other error occur, that error
*              will be displayed and the program will be halted.
*
*
* ---Error 108 = "File in use by another."
IF ERROR() = 108
      RETRY
ELSE
      ? CHR(7)
   ? ERROR(),MESSAGE()
   WAIT
ENDIF
RETURN
* EOP Opt_err
* EOP Opt_proc.PRG
```

```
* Program ...: Opt_req.PRG
* Author ....: J. Mike Hedblom
* Date ......: October 1, 1986
* Note(s) ...: Database file request procedure for Optim.PRG
*
* ---Check for existence of a filename.
* ---If there is no current filename, get one.
IF LEN(TRIM(opt_file_n)) = 0
   DO Opt_file
ENDIF
SET PROCEDURE TO Opt_proc
ON ERROR DO Opt_err
* ---Opt_file.DBF is the list of current file requests.
USE Opt_file EXCLUSIVE
* ---"opt_file_id" is the unique identification code
* ---of this database file request.
opt_file_id = time()
GO TOP
* ---Determine if the requested file has a request in
* ---the list already. If not, open a request and then
* ---RETURN to the main module to execute the request.
LOCATE FOR opt_file_n = filename .AND.;
          opt_file_id <> file_id
* ---If EOF(), then the database file is not in current list.
IF EOF()
   DO Open_req
   RETURN
ENDIF
* ---The file is already in USE.
DO WHILE .NOT. EOF()
   CLEAR
   * ---CLEAR the TYPEAHEAD buffer to avoid unwanted
```

```
* ---erroneous keystrokes.
CLEAR TYPEAHEAD
* ---If the command execution has gotten this far,
* ---there is/was a request in the list. If the
* ---current request is exclusive and does not match
* ---the file in the list, then do ask if the user
* ---will wait.
IF opt_file_r .AND. (opt_file_id <> file_id)
   owait = .Y.
   @ 08,25 SAY STR(wait_state,8) + " waits."
   @ 10,22 SAY "File is in use by another."
   @ 12,25 SAY "Will you wait? Y/N " GET owait
   READ
   * ---If the user will not wait, return to main menu
   IF .NOT. owait
      * ---If a request was made, release it.
      IF wait_state > 0
         DO Close_req
      ENDIF
      * ---Increment wait state so that operation
      * ---will not be perform on database file.
      wait_state = 1
      RETURN
   ENDIF
   * ---If this is the first time waiting, post a
   * ---request and then wait.
   IF wait_state = 0
      DO Open_req
   ENDIF
   DO Opt_wait
ELSE
   * ---The request is not EXCLUSIVE, test for
   * ---EXCLUSIVE USE already?
   IF ex_request
      owait = .Y.
      @ 08,25 SAY STR(wait_state,8) + " waits."
      @ 10,22 SAY "File is in exclusive use."
      @ 12,25 say "Will you wait? Y/N " GET owait
      READ
      * ---If the user will not wait, RETURN to main
      * ---program.
      IF .NOT. owait
         * --- If a request was made, release it.
         IF wait_state > 0
            DO Close_req
         ENDIF
         wait_state = 1
         RETURN
      ENDIF
      * ---If this is the first wait, post a new
      * ---request.
      IF wait_state = 0
         DO Open_req
      ENDIF
      DO Opt_wait
   ELSE
      * ---If there are no EXCLUSIVE requests pending,
      * ---then post the database filename and execute the
      * ---request; otherwise test again for an EXCLUSIVE
      * ---request pending.
      IF EOF()
         DO Open_req
         RETURN
```

```
         ELSE
            LOCATE FOR opt_file_n = filename .AND. file_id
            LOOP
         ENDIF
      ENDIF
   ENDIF
   * ---Since the request database file was closed while waiting,
   * ---reUSE the database file and search it.
   USE Opt_file EXCLUSIVE
   GO TOP
ENDDO
USE
RETURN
* EOP Opt_req.PRG
```

# USAGE TIPS

## dBASE III PLUS

*by Christopher White*

### COPY TO/SET FIELDS

It is possible to COPY TO a new database file from two others using SET RELATION TO and SET FIELDS. Normally, for this operation you would use JOIN or a custom JOIN program, but the following approach is much faster. You can COPY from one-to-one and many-to-one relations. The following commands demonstrate how to do this.

```
USE <File1>
SELECT B
USE <File2> INDEX <Index File>
SELECT A
SET RELATION TO <Key Expression> INTO A
SET FIELDS TO <Fields List>
COPY TO <File3> FIELDS <Fields List from Global Field Pool>
```

For this to work correctly, you need to specify the fields you want from both database files in the FIELDs clause of the COPY command. COPYing without specifying a fields list does not default to the global field pool, but rather to all the fields of the currently SELECTed database file.

## dBASE Administrator

*by David Howlett*

### LOCATE

The LOCATE command stops on locked records regardless of whether they meet the LOCATE condition or not. For example, if two workstations are sharing the same database file and workstation one locks record 4 and workstation two does a LOCATE FOR RECNO() = 10, LOCATE returns the error message "Record is in use by another" to workstation two when the record pointer reaches record 4. The record pointer is then positioned at record 4. The reason this occurs is that LOCATE cannot evaluate the condition for a locked record and therefore generates an error.

One work-around is to set up a error handling routine with ON ERROR DO Err_trap. Your Err_trap.PRG program should contain only the command RETRY. Execute your LOCATE with REST as the scope making sure that you GO TOP before you executing the LOCATE. When LOCATE REST reaches a locked record, control passes to Err_trap.PRG which in turn re-executes the LOCATE REST command. This continues until the locked record is no longer locked and then the LOCATE process continues looking for a match.

Another approach is to lock the database file with FLOCK() before LOCATEing to prevent other users from locking records. When you reach a matching record UNLOCK the database file and execute RLOCK().

# dBASE TOOLS FOR C

*by Robert Boies*

## Supported Compilers

The following compilers are currently supported for compilation of source code files:

1. Version 1.0 of dBASE TOOLS for C

   **LARGE MODEL**
   Lattice C versions 2.15

   **SMALL MODEL**
   Manx Aztec
   C Prime

2. Version 1.3 of dBASE TOOLS for C and the Incentive Disk

   **LARGE MODEL**
   Lattice versions 2.15
   Lattice and Microsoft C versions 3.0

   **SMALL MODEL**
   Manx Aztec
   C Prime

## Future Compiler Support

Other C compilers are presently being considered for future support. Users wishing to see a given compiler certified should notify Ashton-Tate Software Support in writing. Please understand that certification of a given compiler is an intensive process, so preference will be given to compilers with the greatest number of requests for support.

## Graphics Package Certified

The dBASE TOOLS for C Graphics Library has been certified by both Software Support and Software Test. The final certification of the product was performed during the week of July 7, 1986 and was shipped the first week of August 1986. This package provides graphics support for dBASE III PLUS and the Developer's Release in the form of Framework II graph types as well as low-level functions. The package is supplied as an .EXE file only. Programmers who wish to add their own routines to dBASE should use *dBASE TOOLS for C, The Programmer's Library.*

## Use of the Macro

When calling a dBASE TOOLS for C routine, you can use the "&" macro function with a variable in the argument list. You may, for instance, want to use macro variables to construct generalized routines that can perform multiple tasks. For example, the following demonstrates the general syntax.

```
CALL cFUNC WITH "&func_name. &item1.,&item2."
```

If the variables are defined prior to the execution of the CALL, this line of dBASE III PLUS code can be used as a generalized expression for any dBASE TOOLS for C function that passes two arguments. Similar generalized expressions can be developed for command lines with more arguments.

# Framework II

*by Chris Lo Vasco*

## Defining a Key Set for Data Input

The following macro, Keyset.FW2, allows you to specify a set of keys as a mask that is somewhat more flexible than that offered by **@keyfilter**. You can define the number of keys in the input string and the set of accepted keys. You can use it to enter into a spreadsheet or database field cell by first entering the formula edit or label area and then calling Keyset. You can also use it to create forms-oriented data entry in a word frame. You would do this by navigating to a location within the word frame and invoking the Keyset macro with the appropriate parameters.

There are two ways to set up Keyset. You can either set it up as a standard key macro or as a user-defined function. If you do define as a user-defined function, pass it the "key_count" and "key_mask" values as parameters and then call it with a standard macro key. To call it as a user-defined function, use the following syntax:

@Keyset( < length of input string > ," < mask characters > ")

To set up Keyset as a macro, create an empty frame in the Macro section of the Library and name it with the appropriate **Alt-** < key > name. To set up Keyset as a user-defined function, create the empty frame in the User-defined functions section of the Library and name it Keyset. Then, as before, **EDIT FORMULA (F2)**, **ZOOM (F9)**, and enter the formula replacing the assignments to "key_count" and "key_mask" with **@item1** and **@item2** respectively.

```
; Keyset.FW2
;
;  by Christopher White
;  October 1, 1986
;
@local(count,key,key_count,key_mask),
key_count := 30,
key_mask  := "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
key       := "",
@while(count <= key_count,
     key := @keyname(@nextkey),
     @if(key = "{Esc}",@return(0)),
     @if(@not(@isna(@scan(key, key_mask))),
          @list(
              @performkeys(key),
              count := count + 1
           ),
           @beep
     )
)
```

## Engineering Notation

The Software Support Center has received several requests for the ability to format the results of a numeric expression in scientific notation but with the mantissa as a value less than 1. The following program, Engsci.FW2, allows you to do this.

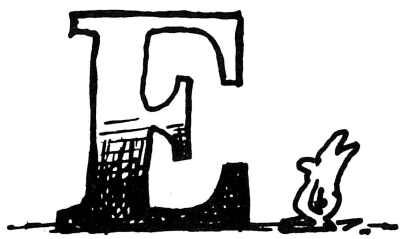
6.023E + 23 <————Scientific notation for Avogadro's number.

.6023E + 24 <————Engsci's notation for Avogadro's number.

The syntax for Engsci is as follows:

@Engsci( < numeric expression > , < decimal places > )

Engsci passes the numeric expression to **@scientific** which formats and rounds the first parameter to the number of decimal places specified by the second. Like **@scientific**, it returns character type. To perform numeric calculations on the results of Engsci, use **@value** to convert the formatted string to a numeric value.



To set up Engsci, create an empty frame in the User-defined functions section of your Library. Press **EDIT FORMULA (F2)** and **ZOOM (F9)** and enter the Ensci.FW2 formula.

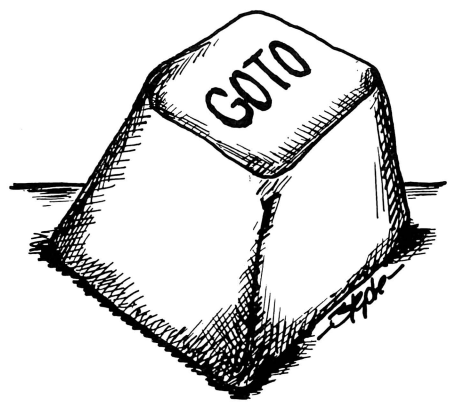
```
; Engsci.FW2
;
;   by Ken Getz
;   October 1, 1986
;
; Declare local variables.
@local(dp, work_val, mantissa, fraction, exponent,
       new_exponent, decimal_pos, exponent_pos),

; Convert to scientific notation, using two as the default
; number of decimal places.
dp := @if(@iserr(@item2),2,@item2),
; If the number passed is not a number, return the value.
work_val := @if(@isnumeric(@item1),
                @scientific(@item1,dp),
                @return(@item1)
            ),

; Find locations of notational parameters.
decimal_pos  := @scan(".", work_val),
exponent_pos := @scan("E", work_val),
mantissa     := @mid(work_val, 1, decimal_pos - 1),
exponent     := @mid(work_val, exponent_pos + 1, 3),
fraction     := @mid(work_val, decimal_pos + 1,
                             exponent_pos - decimal_pos - 1
                ),
; Increase exponent by one.
new_exponent := @value(exponent) + 1,
; Build and send back correct string value.
@return("." & mantissa & fraction & "E" &
        @if(new_exponent >= 0,
            "+",
            "-"
        ) &
```

```
        @integer(new_exponent)
)
```

## Macro to Simulate "Goto Record Number" in a Database

The following macro can be installed into your Library and used to position the cursor on a specified record in a database. The macro calculates the shortest route to a specified row/record and then executes the proper number of **PgUp**s and **Uparrow**s or **PgDn**s and **Dnarrow**s to position the cursor on that row/record.

To install this macro, create an empty word frame in the Macros section of the Library and name it with the **Alt**- < key > you want to invoke it with. Then enter the following formula in the formula section of the frame.




```
; Go_recno.FW2
;
;   by Chris Lo Vasco
;   October 1, 1986
;
@local(cnt, hp, size, power, h, func, func1),
@performkeys("{Uplevel}{F4}"),

; Record the display size of the frame (-2 compensates for
; field labels and line).
size := @panel2 - 2,
@performkeys("{Esc}{Dnlevel}"),
hp := @panel1,
h := @value(@inputline("Enter record number",
                       "0",#TRUE,#TRUE,#TRUE)),
; Change to Table View.
@while(@frameview <> 1,
@performkeys("{F10}")
),

; Decide whether to start from top to bottom.
@if(h <= hp/2,
    @list(
          @performkeys("{Ctrl-PgUp}{Dnarrow}"),
          func := "@performkeys(""""&"{Dnarrow}"&"""")",
          func1 := "@performkeys(""""&"{PgDn}"&"""")"
    ),
    @list(
          @performkeys("{Ctrl-PgDn}"),
```

```
            func := "@performkeys(""""&"{Uparrow}"&"""")",
            func1 := "@performkeys(""""&"{PgUp}"&"""")",
            h := hp - h + 1
        )
),
; For minimum screen activity, remove the semicolon on the
; next line.
;@echo(#OFF),
; Calculate the number of PgUps or PgDns.
power := @int(h/size),
@if(h < size,
    h := h - (power * size) - 1,
    h := h - (power * size)
),
; Execute PgUps or PgDns.
cnt := 1,
@while(cnt <= power,
       @func1,
       cnt := cnt + 1
),
; Execute Uparrows or Dnarrows.
cnt := 1,
@while(cnt <= h,
       @Func,
       cnt := cnt+1
),
@echo(#ON)
```

Do not use **@echo**(#OFF) on any line before the area flagged for the **@echo**(#OFF) in the program above. If **@echo**(#OFF) is placed above this line, the cursor may not be positioned on the proper row/record because of an anomaly with **@performkeys** of **PgUp** and **PgDn** and **@echo** reported in this issue.

## Variable Line Spacing Within a Single Frame

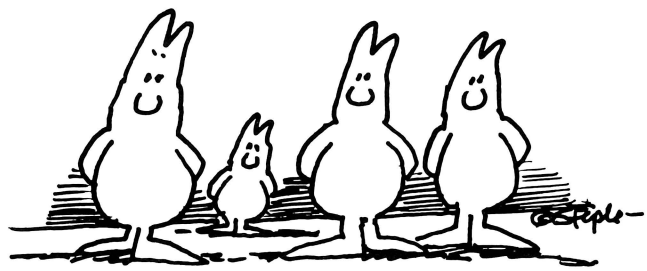
The following is a macro written primarily for law firms, but it can be used by anyone who needs variable line spacing within a single word frame.

Suppose that you need a single word frame to switch from single spacing to double spacing and then back to single spacing. Framework II does not accommodate this directly. Changing the spacing via the **Print:Format Options:Spacing** menu option will make the change for the entire frame and not just for a specified paragraph.

One work-around is to place the text into three separate frames. This would allow you to use the **Print:Format Options:Spacing** to set the proper line spacing in each of the three separate frames. The drawback to this method is the frame borders generate a linefeed unless the three frames are within a containing frame and the subframes have **Frames:Display Labels** off and **Frames:Hide Borders** on. This is a viable option but requires some advanced knowledge of Framework II.

The other method of getting the variable line spacing is to use the following macro, Spacing.FW2. This macro requires that you place the flag ".SP," followed by a number representing the line spacing on a line by itself wherever you want the spacing of the document to change. For example, suppose you have the following text and you want the first three lines single spaced, the second three lines double spaced, and the last three lines single spaced. To prepare this text for use by Spacing, enter the text as follows:

> This is a test of single spacing. This should be printed and displayed in single spacing.
>
> .SP2 ←—— Spacing flag
> This is a test of double spacing. This should be printed and displayed in
>
> double spacing.
>
> .SP1 ←—— Spacing flag
> This is a test of single spacing. This should be printed and displayed in single spacing.



After setting up your document with the appropriate spacing flags, invoke the Spacing macro to reformat. The macro must be invoked from either the frame border or from within the text portion of the frame. Spacing works by looking at each line following the first occurrence of the characters ".SP." When it finds an occurrence of ".SP," it takes the value of the number that immediately follows those characters and spaces the text until another ".SP" is found. To accomplish the proper spacing, it simply inserts the proper number of carriage returns between the lines. The result is a reformatted frame that looks like the following:

> This is a test of single spacing. This should be printed and displayed in single spacing.
>
> This is a test of double spacing. This should be printed
>
> and displayed in double spacing.
>
> This is a test of single spacing. This should be printed and displayed in single spacing.

The frame is now reformatted and will remain this way until changes are made. If you want to make changes to spacing after formatting with the Spacing macro, you can reformat manually or use the macro again with just the portion of text that requires reformatting by inserting ".SPn" where a change is to be made.

To install this macro, create an empty word frame in the Macros section of the Library and name it with the **Alt**-< key > sequence that you want to use to invoke it. Then, enter the following formula into the formula section of the macro frame.

```
; Spacing.FW2
;
;   by Chris Lo Vasco
;   October 1, 1986
;
; Formats flagged paragraphs with user-defined
; line spacing.
;
@local(spacing, lines, lastline, crerror),
@if(@or(@frametype = 9, @frametype = 14),
     @list(
          @if(@frametype = 9,
              @performkeys("{Dnlevel}")
          ),
          @performkeys("{Ctrl-End}"),
          lastline := @panel2,
          @performkeys("{Ctrl-Home}{F6}{Rightarrow}" &
                       "{Rightarrow}{Return}"),
          @if(@not(@textselection = ".sp"),
              @performkeys("{Ctrl-L}S.sp{Return}{Esc}")
          ),
          @while(@and(@textselection = ".sp",@panel2 < lastline),
                 crerror := @panel2,
                 @performkeys("{Rightarrow}"),
                 spacing := @value(@textselection) - 1,
                 @if(@panel2 = 1,
                 @list(
                      @performkeys("{End}{F6}{Home}{Del}"),
                      lastline := lastline - 1
                 ),
                 @performkeys("{End}{Leftarrow}{F6}" &
                              "{Home}{Del}{Dnarrow}")
           ),
           @if(@panel2 < crerror,
               @performkeys("{Return}")
           ),
           @while(@and(@textselection <> ".sp",
                       @panel2 < lastline
                 ),
                 lines := 1,
                 @performkeys("{End}"),
                 @while(lines <= spacing,
                       @performkeys("{Return}"),
                       lines := lines + 1
                 ),
                 @performkeys("{Dnarrow}{Home}"),
                 @while(@isbop(@textselection),
                       @performkeys("{Dnarrow}")
                 ),
                 @performkeys("{F6}{Rightarrow}"&
                              "{Rightarrow}{Return}"),
                 lastline := lastline + spacing
           )
        )
     )
),
@performkeys("{Ctrl-Home}")
```

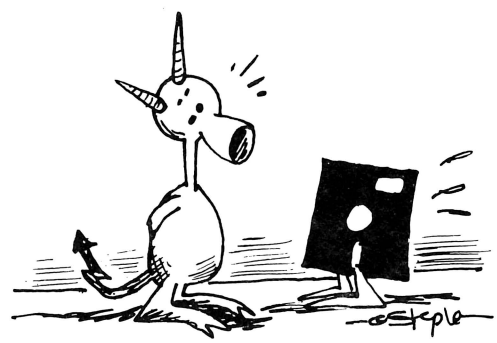
# ANOMALIES

*DEFINITION OF ANOMALY*

*We use the word "anomaly" to mean any one of the following:*

- *INCORRECT COMMAND (or function) IMPLEMENTATION. A command or function does not execute as specified.*
- *INCOMPLETE COMMAND (or function) IMPLEMENTATION. A command or function does not execute as specified in every use of the command or function.*
- *DATA OR LOGICAL BOUNDARY LIMITATION. A command or function cannot perform an operation at a specified data or logical boundary.*

  *NOTE: Ashton-Tate reserves the right to withhold any further information that may reveal proprietary or trade secret information.*

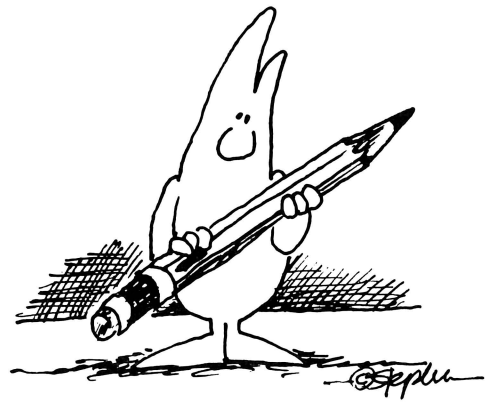


## dBASE Administrator

### Documentation Error

Page 8-17 of the *Learning and Using dBASE III PLUS* manual states that the error message "Unable to Skip" is returned if you attempt to SKIP to a record that is locked by another user.

This is not correct. The error message is returned only when you navigate to a locked record in a full-screen command with **PgUp** or **PgDn**, not as a result of SKIP.



## First Logical Record of a Database File

The first record of a database file does not properly generate a "Record in use by another" error message when locked. For example, two workstations are using the same database file. Workstation one locks record 1. Workstation two then executes the command LOCATE FOR RECNO() = 10. This should result in the error message "Record in use by another" when the record pointer reaches record 1. Instead the LOCATE ignores that record and positions itself at record 10.

There is no work-around.

## RLOCK() and RUN/!

RLOCK() is designed to lock a record, allowing only one user at a time to make changes. If you lock a record with RLOCK(), any attempt by another workstation to use that record should return the error message "Record is in use by another." However, if a workstation locks a record with RLOCK() and subsequently issues the RUN/! command, the record is unlocked though DISPLAY STATUS at the first workstation still shows the record as locked. In addition, the record is available to another user who can also RLOCK() and EDIT it.

At this point both users appear to have locked the record and both are able to EDIT the record at the same time. The actual changes to the record, however, depend on the last user to have control. The only way to ensure data integrity is to use the UNLOCK command before issuing a RUN/!.

## SKIP + <n>

If you issue a SKIP + <n> and a record is locked between the current location and the destination, the error message "Record is in use by another" is returned and the record pointer is positioned at the locked record. If the database file is indexed, you cannot determine how many records were SKIPped before you encountered the locked record. To work around, use the following commands in place of SKIP + <n>.

```
SET CONSOLE OFF
LIST NEXT <n>
SET CONSOLE ON
```

# dBASE TOOLS FOR C

## Character Strings (Version 1.0)

Since the version 1.0 allows you to use spaces in addition to commas as delimiters in the command line, any attempt to pass a string that has embedded spaces results in the compression of the spaces.

This is fixed in version 1.3 and the Incentive Disk.

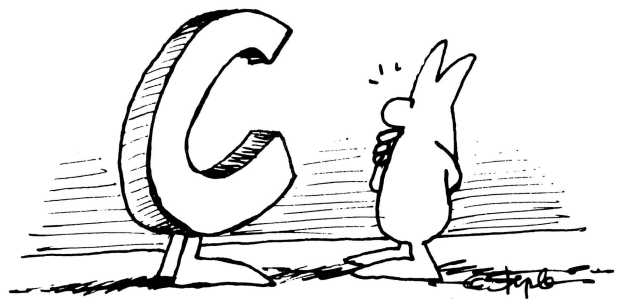
## dBASE III PLUS Memory Variables (Version 1.0)

On occasions dBASE TOOLS for C arrays may overwrite dBASE III PLUS memory variables or you may find garbage characters in your arrays. This problem may occur for two reasons. If you use an array space greater than 256K or if HISTORY operates concurrently with DCT1.EXE. You can work around this problem in version 1.0 by eliminating HISTORY from the environment. This is accomplished by adding the following lines to your CONFIG.DB file.

```
HISTORY = 0
HISTORY = OFF
```

You should also maintain array space at less than 256K including DCT1.EXE which on the distribution disk is 70K.

The problem has been corrected in version 1.3.



## Loss of Available Memory (Version 1.0)

If you load dBASE TOOLS for C version 1.0 from within dBASE III PLUS with the RUN command, QUITting dBASE III PLUS under some circumstances results in the DOS error message "Memory Allocation Error, Cannot Load COMMAND.COM, System Halted." The only way to recover from this error condition is to turn off your computer and cold boot. The problem is not specific to dBASE TOOLS for C, and can happen any time you load a TSR program from a processor unable to behave as a substitute for COMMAND.COM.

To work around, RUN COMMAND first instead of RUNning DCT1.EXE. Then, from the DOS prompt, load DCT1.EXE. When you return to the DOS prompt, EXIT the command shell to return to dBASE III PLUS.

This problem does not occur in the Developer's Release. In addition, it is fixed for use with dBASE III PLUS in the version 1.3 and the Incentive Disk.

## Microsoft and Lattice Versions 3.0

When Microsoft and Lattice released version 3.0 of their respective C compilers, the internals of the compilers and libraries changed significantly. As a result of these changes neither compiler is supported beyond version 2.15 with dBASE TOOLS for C version 1.0. You can, however, use version 3.0 of the either compiler with the respective 2.15 versions of the compiler libraries.

# Framework II

## @performkeys and @echo(#OFF)

**@performkeys** of the **PgUp** and **PgDn** keys may not work correctly when preceded by **@echo**(#OFF). This appears to happen because some of the status pointers are not updated properly with **@echo**(#OFF).

## EXTEND SELECT (F6)

If you delete subframes of a containing frame by **EXTEND SELECT (F6)**, **Dnlevel**, and then **Del** from the parent frame, the computer may freeze requiring a system reboot.

The problem only occurs when you do this from the fourth or subsequent section, or from the fourth or subsequent levels of any section containing three or fewer subframes.

The work-around is to press **Dnlevel** from the parent frame before you **EXTEND SELECT (F6)** and **Del** to delete subframes.

## Importing Lotus 1-2-3

When the Lotus 1-2-3 function **@iserr**() is imported into Framework II, the left parenthesis is omitted. This results in the error message "Undefined function reference" when a cell containing the imported formula is recalculated. For example, if the 1-2-3 worksheet contains the formula **@iserr**(a1), Framework II imports it as "**@iserr**a1)".

The work-around is to edit the formula and insert the parenthesis after the last "r" in iserr.

## Using @print with a Local Variable

If you use a local variable as the **@print** function argument, your computer will freeze, requiring a system reboot. However, the **@print** function works properly if you use global variable or frame as the argument instead. For example,

```
; This will not work.
@local(variable)
variable := "This is a test."
@print(variable)
```

# CORRIGENDA

In the August 1986 issue of ***TechNotes***, on page 7, Table 1 in the article "Framework II Printer Drivers," lists the LQ1500 printer driver filenames incorrectly. For standard resolution graphs, the printer driver filename is LQ1500_1.PR instead of LQ1500_2.PR. For high resolution graphs, the printer driver filename is LQ1500_2.PR instead of LQ1500.PR. As noted in the table, the standard resolution printer driver (LQ1500_1.PR) is available through the Ashton-Tate Software Support Center or the Ashton-Tate Forum on CompuServe.

# ON-LINE SERVICES

Do you have a modem? Do you subscribe to CompuServe or The Source? If so, we have great news for you! Ashton-Tate offers technical support on all of these electronic information services. You can quickly access the kind of quality technical information that you have come to expect from ***TechNotes***, and you can do it through your computer system. To take advantage of this progressive new support method, you must subscribe to one of the services listed below. There are no additional charges for using our on-line support services, apart from the normal connect-time fees for each service. You can get specific subscription information by calling one of the following numbers:

| Service | Subscriber Information | Command to find Ashton-Tate |
|---|---|---|
| CompuServe | (800) 848-8199 | GO ASHTON |
| The Source | (800) 336-3366 | ATSIG |

To give you an idea of the fast-moving dynamic support that is happening on-line, ***TechNotes*** will be publishing regular excerpts from the Ashton-Tate Forum on CompuServe. This monthly material comes from messages our Forum users have posted on the bulletin board. To retain the original sense of this material, we have edited only for spelling, and all message headers have been removed to protect the identities of our users. Every other message within a single conversation topic has been italicized to enhance readability.

---

## dBASE II, dBASE III, dBASE III PLUS:

I am working with dBASE II version 2.43*, and I am trying to send a custom report to disk, which I can do with the SET ALTERNATE command. My question is how can I then print out that report, whether it be custom made or a REPORT FORM once it is on a disk?

*You can print a disk file from DOS by a number of methods, such as using the PRINT.COM program that comes as part of DOS, or by COPYing the file to the printer with a DOS command, such as COPY <filename> > PRN:.*

Okay, that covers printing the file after leaving dBASE II. Do you know of any way, while still in dBASE II, to do this? I would prefer to have the operator be able to give all of her commands while within the program itself, and I understand you can't execute DOS commands while still in dBASE II.

*dBASE II does not have this capability. About the best you could do is to use the QUIT TO command to branch out of dBASE II to do the output, then in that QUIT TO command include a call to return to dBASE II. dBASE III, however, does have the capability to issue DOS commands without leaving dBASE, via the RUN command.*

.....

When I type "DBASE < Filename > " at the DOS prompt, dBASE III PLUS displays the copyright notice and requires the user press **Return** before the Filename.PRG file executes. I need to prevent the system from waiting for operator input at that stage. I have long batch files intermixed with dBASE III PLUS and other programs, all of which run unattended. If human response is required, we can't run them. How do I get around this problem?

*There's no need to press* **Return**, *or use any patch or utility. dBASE III PLUS will automatically continue past the license agreement after a certain time period, depending on your machine. A basic PC will time-out after about 10 seconds, an AT may have no significant delay. An unattended setup, like you propose, should require no extra effort on your part.*

.....

I'm relatively new to programming, although I've used dBASE II for a couple of years and now program occasionally in both dBASE II and dBASE III. I've run into a problem I hope you can solve for me. Using dBASE III, I have two files open in areas 1 and 2, and am using a DO WHILE loop to store fields from the first file into memory variables, then SELECTing 2 and REPLACEing fields with the stored data (some numeric, some character). I received the message, "Variable not found," so I tried declaring the variables PUBLIC before the loop began so I could check them after running the program. This time I got a "Data type mismatch" message, and when I checked the memory variables, they were all logical type, containing a value of false (.F.) I eventually got the program to run after removing the PUBLIC statement, but can anyone tell me why this happened?

*When you declare a memory variable as PUBLIC, dBASE III sets it up as a logical field type. It is necessary to initialize the variable to whatever type is required on the next line of code if you don't want a logical memory variable. If you want a character type memory variable and want it to remain blank until your routine uses it, you can do the following:*

```
PUBLIC mvar
STORE SPACE(1) TO mvar
```

.....

I used CONFIG.DB to customize my function keys with dBASE III but so far have not been able to get them to work in dBASE III PLUS. I get an error message as the program boots up.

*The first part of the mystery is that dBASE III requires no quotes around the function key assignment in the CONFIG.DB file, and dBASE III PLUS does. For example, F8 = MODIFY COMMAND is permitted in dBASE III, but dBASE III PLUS requires F8 = "MODIFY COMMAND ." Second, if you want to build in a* **Return**, *so something like "DO Program" executes when you press the function key, the semicolon must go inside the quote marks, i.e. F8 = "DO Program;".*

We have a problem changing DATE() + 10 days to express a form like English, so that the result of the expression CTOD("06/05/86") + 10 prints as "June 15, 1986."

*Try one of these two methods:*

```
STORE DATE() + 10 TO newdate
STORE CMONTH(newdate) + ;
      STR(DAY(newdate),3) + ", " + ;
      STR(YEAR(newdate),5) TO cnewdate
@ <row>,<col> SAY cnewdate
```

*or*

```
? CMONTH(DATE()+10) + " " +;
  STR(DAY(DATE()+10),2) + ", " +;
  STR(YEAR(DATE()+10),4)
```

Can anyone tell me if it's possible to convert a character string ($1.25) to an integer for calculations?

*Remove the leading character, $ using the SUBSTR() function, then take the VAL() of the remaining string. You can also use the &. Either way works. For example,*

```
cnum = "$1.25"
cnum = SUBSTR(cnum,2)
SET DECIMALS TO 2
nnum = VAL(cnum)          && Result is 1.25.
nnum = &cnum              && Result is 1.25.
```

Does dBASE II work on a Commodore 128 using a 1571 disk drive?. The new Commodore 128 has CP/M capability, but I can't seem to find a copy of dBASE for that machine.

*We don't provide a format specifically for the Commodore. However, that machine and drive can read Kaypro II disks, and dBASE II is supplied in that format. You may have to modify the screen installation options a little bit, but it should run fine.*

---

# Framework II:

I've just started using Framework II and so far I love it! I've put together a simple spreadsheet to handle the monthly mortgage payments we receive. Since we calculate interest daily it is important to be able to calculate the number of days from the last payment. Presently I am calculating the number of days on my HP-12C. But I figure if a hand-held calculator can do it then certainly a PC running Framework II can. I just don't know how to go about doing it.

*If you are using a date cell in the form:* ***@date****(year,month,day), then you can use the* ***@diffdate*** *function like this:*

```
@diffdate(@today,cell)
```

*If your date cell is character, however, you must convert it to a Framework II date. For instance, if your cell A1 contains 06/10/86, you make a date with this formula:*

```
@date(1900 +
        @value(@mid(A1,7,2)),
        @value(@mid(A1,1,2)),
        @value(@mid(A1,4,2))
)
```

.....

I want to be able to use condensed printing for certain portions of a frame, while using regular print for the rest. I have tried using **EXTEND SELECT (F6)** for highlighting those portions and turning the toggle for condensed printing on in the print directory, but then everything prints condensed. I use an Epson JX-80 printer. Any suggestions?

*Selecting **Condensed Print** in the **Print:Format Options** menu sets the attribute of an entire frame. If you want to have a portion of a document print in condensed you have two choices. First, you can split the document into separate word frames and set **Condensed Print** as the attribute of the frame you want and put both frames inside a containing frame. Second, you can embed printer control codes right in a word frame, at exactly the spot where you want to change to condensed. For instance, if you have an Epson printer, you put the cursor in the word frame where you want the condensed to begin, and hold down the **Alt** key and type 15 on the numeric keypad. Release the **Alt** key, and a special character will appear in the frame. Now just print the frame. Be sure to insert a code to turn off condensed when you are done with it. For the Epson, it's an ASCII code 18.*

.....

Have you had any experience with the AST and QuadRam Enhanced Expanded Memory Specification? It got a write-up in January's *PC World* under the title "The Ram Revolution." Because it can have 64 windows of 16K each open at a time, rather than just four like Intel's Above Board, it is supposed to be much better for users of programs like Framework II. Can you verify the article's claim that Ashton-Tate is supporting the EEMS and that Ashton-Tate prefers it over the EMS for use with Ashton-Tate programs?

*We have no official preference. We have used both the Intel Above Board and the AST Rampage! and find them both effective. Framework II supports the Intel Above Board, AST Rampage! and Advantage!, 286 "motherboard" memory, and hard/ram disk spooling.*

.....

I can't get the **Send Control Characters** setting in the **Telecommunications: Setup:Options** menu to change from off to on. What am I doing wrong?

*The **Send Control Characters** option has the unusual habit of closing the menus when it is toggled. If you select it, then reopen the menus, you should see that it has changed to the opposite state.*

.....

I'm having a problem with the outline function when I want to have a paragraph or so of an introduction and then want the regular outline structure to go below the introduction, e.g.:

narrative
.....
.....
.....
1. Outline Points
   1.1 Further Points
2. Second Point
   2.1 More Stuff
      2.1.1 Picky Little Point

I couldn't get the full introduction to print without having the outline expand into the larger format, which didn't look very good. Any suggestions? I tried using more containing frames but they just ended up as blank bullets on the outline! Thanks.

*The problem is that any given frame at the desktop level can be entirely outline view or entirely frame (contents) view. I experimented with this a bit, and found some ways to make it print in the way you want. The most obvious alternative is to take the outline portion, write it out to a text file, load it back to a word frame, and put it in a containing frame with the introduction. If you will be modifying the outline frequently, this may be a little inconvenient. But considering that what you want to do is not available as a direct feature, it's not too bad.*

*The other technique involves a few tricks. Your introduction frame would be the first frame in the outline, and it would have* ***Number Labels*** *off. You print the containing frame in frames view, and control the appearance of the outline portion with various frame attributes. The outline frames should contain no text. The headlines will print double spaced, unless you use* ***Allow Free Dragging****, and arrange them. The headlines will not be indented, so you may want to use* ***Page Offset*** *to simulate some indentation. In other words, count on the frame labels to print like an outline, and touch them up by experimenting with frame attributes.*

.....

When can I expect to see a revised Framework II, where the control key functions work on the new IBM systems with the new keyboards?

*That's a good question, and fortunately, I have a good answer! We have just completed testing of a new keyboard driver for Framework II that is compatible with the XT2 and AT2, as well as all the older models. The keyboard driver is the only thing that needs to be changed for Framework II to work on the XT2/AT2. The driver is available for downloading on both CompuServe and The Source, and is also available on request directly from the Software Support Center. Use the SETUP program in Framework II to install the new driver.*

---

# MultiMate:

I cannot get a NEC 8850 printer equipped with the correct print wheel to print special symbols for paragraph and section, using MultiMate. The printer doesn't seem to recognize ASCII 14 and 15. Can you help?

*To access alternate characters on the NEC 8850, put "02", the hex code for the desired symbol, "03" to go back to regular characters in the CWT (Character Width & Translation) table in the NEC3550.PAT. Refer to your printer manual for the correct hex codes for the paragraph and section symbol.*

.....

# MULTIMATE TECHNICAL SUPPORT

*Submitted by Liz Burke*

*Contributions by Clay Carley, Lisa Chouinard, Bob Gray, John Hornyak, Sue Shafer*

## MultiMate Magic

### "Document Full" Error Message

WHAT IF...you are typing in a document and receive the error message "DOCUMENT FULL ERROR: UNABLE TO SAVE PAGE" or "PAGE FULL: UNABLE TO SAVE PAGE." What does this mean and how do you save your document?

A "DOCUMENT FULL..." or "PAGE FULL..." error message means that you have exceeded one of MultiMate's document limits. Those limits are listed below:

1. The maximum number of characters per page is 6,144.
2. The maximum number of lines per page is 150.
3. The maximum number of pages per document is 250.
4. The maximum size of the entire document is 128K bytes.

Page 61 in the July 1986 issue of ***TechNotes*** includes a chart for determining the maximum number of lines per page based on the length of the format line. To keep the size of the document under 128K bytes, a general guideline to follow is to keep the page count at 50 or below.

If you do receive one of these error messages, you will lose some data. The amount of data lost depends on whether the option "Keep Document Closed for Safety" is set to "Yes" or "No." If it is set to "Yes," MultiMate Advantage will save your data as you progress from one page to the next. That way, if you do get one of the error messages, only the current page is lost. On the other hand, if it is set to "No," MultiMate only saves your work when you press **F10**. So, if you add 20 pages to an existing document before getting the message "DOCUMENT FULL...," all 20 pages of work are lost.

To work around the problem, use the External Copy (**Shift-F8**) to divide the document into two separate files. To do this, follow these steps:

1. Exit the document and return to the main menu. To do this, you may have to try several ways:
   a. Press **F10** to try to save the document.
   b. If you receive the same "DOCUMENT FULL..." error message, press **Alt-3**. If this is successful, it will take you to the "Print Parameters for Document" screen. Do not attempt to print your document at this time. At this screen, press **Esc** to return to the main menu.
   c. If **F10** and **Alt-3** do not work, press **Esc**. Answer "Yes" to the prompt "Do you wish to escape without saving page ?"

2. Once you have returned to the main menu, create a new document.
3. On the first page of the new document press **Shift-F8**.
4. Enter the name of your old document and the drive letter where it resides, then press **F10**. MultiMate then searches the specified drive for the old document and places the cursor on the first character of that document. The prompt "START COPY WHERE?" displays in the upper righthand corner of the screen.
5. Move the cursor to the beginning of the text to copy and press **Shift-F8**. If you also want your format line to be copied, press **F9** now.
6. Highlight the text to copy (about half of the document) and press **Shift-F8** again to complete the copy process.
7. Repeat the above steps to copy the second half of the document to another new file.

## Changing Format Lines

WHAT IF...you have a 20-page document and wish to edit the document format line without having to change the format line on each page?

Follow these steps:

1. At the first page of the document, press **Shift-F6**.
2. You will be prompted for the type of replacement: (1) GLOBAL, (2) DISCRETIONARY, (3) ABORT. If you want to replace the format line on each page, press 1 for GLOBAL. If you want to replace only a few format lines, press 2 for DISCRETIONARY.
3. Press **F9** and edit the format line.
4. Then, press **F10** to replace the format lines in the document. If you chose GLOBAL, MultiMate replaces all of the format lines. If you chose DISCRETIONARY, MultiMate prompts you with "REPLACE? Y/N ANY OTHER KEY TO ABORT" at each format line in the document. Type "Y" to replace the format line or "N" to go on to the next one.

For additional information, see the "Format-Replace" section of your *MultiMate Reference Manual.*

## Saving Bytes when Copying

WHAT IF...you create new documents by editing copies of old documents and find that, even when edited, the new documents occupy the same amount of disk space as the original documents?

You can reduce the size of the new document by using the External Copy function. To do this, follow the steps below:

1. From the MultiMate Main Menu, create the new document.
2. From page 1, press **Shift-F8** to move to the old document.
3. Enter the name of the old document and the drive letter where it resides, then press **F10**.
3. Once you are in the old document, move the cursor to the beginning of the text to copy, and press **Shift-F8**. The screen prompt "COPY WHAT?" will display. Highlight only the text that you want to bring into the new document.
4. Then, press **Shift-F8** again to execute the Copy function.

## MultiMate On-File Names

WHAT IF...you are unable to copy your MultiMate On-File card box through DOS?

Make sure that the On-File card box filename does not contain a space; for example, Mailing List or New Employees. If your filename requires more than one word, use a dash or underscore instead of a space to separate the words. An example of this is Mailing-List or New_Employees.

If you have already created an On-File card box filename that contains a space, rename all three associated files (.IDX, .CRD, and .TEM) through BASICA, which is located on your DOS system disk. To do this, follow the steps below:

1. Load BASICA.
2. Use the following syntax to change the filenames:

   NAME " < drive > < Filename.EXT > " AS
   " < drive > < Filename.EXT > "

   For example:

   NAME "C:Mailing List.TEM" AS "C:Mailing-List.TEM"

   The "Ok" prompt indicates that the file has been renamed.
3. Repeat step 2 for each filename.
4. To exit BASICA, type SYSTEM.

## Using the ASCII Conversion Utility, UTIL

MultiMate's conversion utility, UTIL, converts ASCII text files to MultiMate document files and vice versa. This utility allows you to share data with other software programs. Below are the formatting choices from the UTIL menu and an explanation of what each option does.

## To ASCII From MultiMate Advantage:

Use Document Summary Screen for Page One? YES NO

If you want your "Document Summary Screen" to be the first page of your ASCII file, choose YES. If you don't, choose NO.

Accept Extended Characters? YES NO

If you choose YES, MultiMate Advantage will include in the ASCII text file any extended characters (ASCII 127 - 255) that are in your document. If you choose NO, they are not.

Line End Sequence? CR LF LF CR

This option defines what MultiMate Advantage puts at the end of a line. Most printers recognize CR/LF (carriage return/line feed) as a line terminator. If, however, you are going to send the text file to a TTY device, choose CR (carriage return). LF (line feed) is used mainly when you wish to transfer data to mainframe computers.

Page End Sequence? blank line Form feed No page end

This option defines what MultiMate sends out at the end of each page. Some software programs use a form feed character to separate pages. Others need exactly 66 lines on every page, and still others do not accept blank lines or form feed characters at all.

If you choose form feed, the form feed character, 0C hex, is placed at the end of each page. If you choose blank lines, blank lines are sent from the end of the text to the end of the page. The number of blank lines sent is determined by the document page length minus the number of lines of text. For example, assume the document page length is set to 66 lines per page, and the text takes up 60 lines. In order to reach the limit of 66 lines per page, six blank lines are sent to the ASCII text file. If you choose no page end, nothing is placed at the end of each page.

Replace Tabs with Spaces? YES NO

This option will affect the format of your file. If you answer YES, each tab character is replaced with eight spaces. If you answer NO, a tab character is sent to the text file. Choose YES unless you are using a software program that has tab stop setting capabilities.

Remove Underline/Doublestrike Sequences? YES NO

If you choose YES, the bold, strikeout, and underline attributes are not written to the ASCII text file. If you answer NO, the codes used for bold, strikeout, and underlining are included in the ASCII text file. For example, if you choose NO, the word **Test** in bold looks like this (^H is the control code for backspace):

t^Hte^Hes^Hst^Ht

If you are going to print the file on a printer that does backspaces, choose NO; otherwise, choose YES.

## To MultiMate From ASCII:

Remove Page One? YES NO

If you want to delete the first page of your ASCII file, answer YES; otherwise, answer NO. If you first converted the file to the ASCII format through UTIL, and you answered YES to "Use Document Summary Screen as Page One," then you may want to answer YES to this option. This way the "Document Summary Screen" is not the first page of your document.

Remove Extended Characters? YES NO

If you choose YES, all extended characters in your ASCII file are converted to periods when the file is brought into MultiMate. If you choose NO, MultiMate will try to display them as they are. The MultiMate tab character and the MultiMate return character always translate to a **Tab** and **Return**, respectively. Any character that has an ASCII decimal value less than 32 is converted to the MultiMate control code format **Alt**-a000, where 000 is the decimal value of the character. A character that has an ASCII decimal value of 127 or greater displays as that character. Merge code symbols and required page break symbols display as line and box symbols.

Line End Sequence? CR/LF LF CR

This option tells MultiMate how each line terminates. If you use the DOS TYPE command to view the text file and it looks fine, the line terminator is probably CR/LF (carriage return/line feed). If the text prints out as one continuous line, it's probably CR (carriage return). If the text prints on more than one line, but does not wrap to the left margin, then it's probably LF (line feed).

Remove **Returns** Where Possible? YES NO

This option minimizes **Returns** in the MultiMate document. If you choose YES, MultiMate inserts a format line when the length of a line varies by five characters from previous line lengths, thereby minimizing **Returns**. If you choose NO, MultiMate sets the format line to the longest line on the page and then puts a **Return** after each line of text.

Lines Per Page? (1-150) 055

This option determines the number of lines per page and therefore where the page break should be. However, regardless of the line count, if MultiMate sees a form feed character (0C hex) in the text file, it will insert a page break at that point.

Format Line
F9 to Edit Format Line

If you choose to edit the format line, you can change the line spacing and setup tab stops for MultiMate to follow as it's converting the ASCII text file to a MultiMate document. If the ASCII text file contains tab characters, they will convert according to the tab stops in this format line.

This format line only affects the line spacing and tab stops, assuming your text file contains tab characters. The format of the document as a whole is determined by the "Remove Returns Where Possible" option.

# INTERNATIONAL SUPPORT

*by Paul Bukauskas*

## A Helpful File at Print Time: DBPRINT.PTB

This month's International Support section features DBPRINT.PTB, a file which is available on all international releases of dBASE III and dBASE III PLUS. This file provides printer support for international characters. Since international characters vary from one language to the next, each foreign language version of dBASE III and dBASE III PLUS comes with its own distinct DBPRINT.PTB file. It defaults to Epson printers, a setting that cannot be altered under dBASE III version 1.1. However, dBASE III PLUS allows you to customize this file for your printer.

Even though an extended character appears correctly on the screen, it may not print as the same character. For example, on an Epson FX-80 printer, ASCII 165 appears on the screen as "N" but prints as "Ø". The character prints correctly only when the printer manufacturer uses the same ASCII value as used by the computer. IBM attempted to standardize these codes when they introduced the IBM extended character set, but not all printer manufacturers chose to follow it.

Using the default Epson mode, let's look at how it works. The international versions of dBASE III and dBASE III PLUS look for the DBPRINT.PTB file when loading. If it is found, dBASE brings the contents into a 512K-byte buffer area. Then, each time dBASE makes a print request, this buffer is consulted in order to substitute a value that the printer does not support. Although the printer's internal ROM is not changed, dBASE III captures the value and reinterprets it.

To illustrate the point, let's assume you want to print the character "U" (ASCII 237). Since an Epson FX-100 will normally print an italicized "*m*" when it receives the ASCII value 237, DBPRINT.PTB sends the following command sequence:

1. Print a lower case "o", (93H, ASCII 097).
2. Send a backspace command to the printer, (08H, ASCII 008).
3. Print a backslash "\", (2FH, ASCII 047).

### Customizing DBPRINT.PTB

The international versions of dBASE III PLUS allow you to build custom DBPRINT.PTB files. A menu-driven utility program called Genptb prompts you for a replacement sequence for characters that you want to change.

For example, suppose your printer does not support the character (93H). Genptb prompts you to modify DBPRINT.PTB so your printer can print this character. The codes that you would enter are italicized:

CODE: *93H* **Return**
REPLACEMENT BYTES: *'o',08H,60H* **Return**

where 93H is the hexadecimal value of ASCII 147 (ô), and the replacement codes instruct the printer to print "o", backspace, and print a circumflex.

## When Not to Use DBPRINT.PTB

We've seen how DBPRINT.PTB can be used to access characters which printers do not directly support. Problems arise, however, if DBPRINT.PTB is used when it shouldn't be. This is the case if your printer does support an extended character set and you copy DBPRINT.PTB to the dBASE directory anyway. The table below illustrates this problem. The characters were printed using an IBM Graphics printer under the Spanish version of DOS 2.1, dBASE III, and DBPRINT.PTB.

The column on the left contains characters printed directly from a dBASE III screen using **Shift-PrtSc** and the column on the right are the same characters printed from dBASE III using SET PRINT ON. As you can see, DBPRINT.PTB has reinterpreted the characters that were meant for the printer. Had DBPRINT.PTB not been loaded, the characters would have been passed to the printer correctly.

| **Printed Character** | |
|---|---|
| **Shift-PrtSc** | **SET PRINT ON** |
| i | i |
| ¿ | ] |
| N | \ |
| á | a |
| é | e |
| ĭ | i |
| ó | o |
| ú | u |
| Ü | }{ |

In summary, DBPRINT.PTB is a useful tool for printing extended characters on printers that do not support them. If your printer does print extended characters, do not copy DBPRINT.PTB to the dBASE subdirectory.

# Tips for International Versions

## ASCII 161

If you are using the U.S. release of dBASE III PLUS and you place the ASCII character 161 (ĩ) into a database or format file, dBASE III and dBASE III PLUS will strip the eighth bit. This results in a carriage return (HEX 0D) rather than the original character. This does not occur when you use an international version of dBASE III and dBASE III PLUS.

## Using the Spell Checker

The U.S. version of Framework II does not support extended characters during a spelling check. If you add a word to the dictionary that contains extended characters, only the letters up to but not including the extended character are added. For example, if you run the spelling checker on the Spanish word Manana, the letters "Ma" are highlighted as a misspelling. If you attempt add the word to the dictionary, only "Ma" is added. The rest of the word is left out.

International versions correctly handle extended characters during a spelling check.

# LETTERS

## Indexing by Week

Dear ***TechNotes***:

I am a dBASE III user. One of my database files, Sell.DBF is a transaction file that tracks what items are sold on what day of the week. To make the database file more accessible, I have INDEXed it ON Selldate, the date the item was sold. I have no problem using the REPORT FORM to get my totals, but I need a report with weekly subtotals. Can you give me some help explaining how I can do this?

C.W.

Dear C.W.:

The REPORT FORM has an option for grouping on a field or expression. This is the only way to get subtotals using the REPORT FORM. If you group on Selldate, then the result would be a total for each day. So to group by week, you can use the following expression.

```
Selldate - DOW(Selldate)
```

The DOW() function returns a value that corresponds to the position of day of the week. Sunday is 1 and Saturday is 7. The only problem here is that the group subtotal gives you the Saturday that precedes the Selldate, so add one to the expression to give the correct date. Therefore, to group by week with first day of the week on Sunday, use the following expression.

```
Selldate - DOW(Selldate) + 1
```

For the group heading, you can say "FOR WEEK OF: ".

## Running Programs in Framework

Dear ***TechNotes***:

One of my FRED programs allows the user to choose a list of files to run through DOS Access from a menu. However, the user wants to add more files to the menu on a regular basis. Is there a way to have the user type in a list of commands and have them execute in DOS?

C.M.

Dear C.M.:

The following program, Batch.FW2, uses a combination of **@inputline** and **@run** to batch a series of DOS commands together so you can execute them from the desktop as a macro. The trick to this program is having the user input a string of DOS commands separated by semicolons, writing a .BAT batch file, and then executing it. As an added feature, the **@setdirectory** command is used to return to the current subdirectory after execution of the .BAT file.

To set up Batch.FW2, create an empty word frame in the Macros section of the Library and name it with the **Alt**-< key > you want to invoke it with. Then enter the following formula in the formula area of the frame. The results of the batched DOS commands are returned into a frame called Dos as long as a called DOS program doesn't clear the screen.

```
; Batch.FW2
;
;   by Doug Haman
;   October 1, 1986
;
; Set up local variables.
@local(temp, comm, current),
; Set abnormal end to #FALSE.
@isabend(#FALSE),
; Capture location of COMMAND.COM.
comm := @getenv("COMSPEC") & " /C ",
; Capture the current subdirectory.
current := @setdirectory,
; Create an empty word frame and label it Dos.
@performkeys("{Ctrl-C}EDos{Return}"),
; Check the status of Locate:Contents Included.
@if(@not(@sense(#LC)),@performkeys("{Ctrl-L}C{Esc}")),
; Display all keyboard input.
@echo(#ON),
; Display prompt and input keystrokes.
temp := @inputline("Enter DOS command - separate " &
                   "commands with a ;"),
; Put new commands into Dos.
Dos := temp,
; This is a loop to remove all semicolons and put each command
; on its own line.
@while(@not(@isabend),
         @performkeys("{Ctrl-L}S;{Return}{Esc}"),
         @if(@not(@isabend),
        @performkeys("{Del}{Return}")
    ),
),
; Move to frame border.
@performkeys("{Uplevel}"),
; Write DOS.BAT file.
@writetextfile("DOS.BAT",Dos),
; Run DOS.BAT file.
@run(Dos,comm & "DOS.BAT"),
; When finished with DOS.BAT, return to the current
; subdirectory.
@setdirectory(current)
```

# Transferring One Record Between Database Files

Dear ***TechNotes:***

I am a user of both dBASE II and dBASE III. One of my programs requires two database files, Master.DBF and Trans.DBF in USE at one time. My program checks every transaction to make sure a master code exists in Master.DBF. If the code does not exist, then I need to move information from that specific record in Trans.DBF into Master.DBF. Until recently, I STOREd Code TO mcode, closed Trans.DBF, SELECTed the work area where Master.DBF exists, and APPENDed FROM Trans FOR Code = mcode. Although this took forever, it did work. Now, I have learned to leave both Master.DBF and Trans.DBF open, SELECT Master, APPEND BLANK, and REPLACE all pertinent fields from the current record in Trans. Although this is much quicker, is there a quicker way?

E.S.

Dear E.S.:

Depending on the number of fields you are transferring, it may be quicker to COPY NEXT 1 TO Temp from Trans, SELECT Master, and APPEND FROM Temp. This only transfers information from Temp with the same field names in Master. If however, there are specific fields that you don't want filled, then the current method you are using is the quickest.

# READER SUBMISSION FORM

If you have a programming tip, a handy formula, or a short utility, and you are interested in sharing it with other Ashton-Tate or MultiMate enthusiasts in a future issue of ***TechNotes***, please read the provisions below. After reading these guidelines, fill out this form and send it along with your submission(s) to the following address. Submissions must be made on disk (IBM or compatible format) in standard ASCII format or in Framework II (.FW2) format.

***TechNotes* Submissions**
**Software Support Center**
**Ashton-Tate**
**20101 Hamilton Ave.**
**Torrance, CA 90502-1319**

NAME _____ DATE _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE (During normal business hours) (_____) _____ - _____

_____

**(SIGN HERE TO ACKNOWLEDGE YOUR UNDERSTANDING OF THE PROVISIONS BELOW)**

To establish a clear understanding of the rights and usage of submissions, you are asked to first represent and warrant that:

1. You are the author of the material: that it is original and has not been heretofore published; that it does not infringe upon any copyright, proprietary right or any other right whatsoever; that it contains no matter that is contrary to law; and that you are the sole owner of the rights herein conveyed to us, and that such rights are free and clear.
2. We shall have the right to use your name, biography or other information pertinent to your submission.
3. We shall have the right, at our discretion, to edit the material to suit the requirements of our publication.
4. Suggestions, comments, and programming materials submitted to ***TechNotes*** are not for reply. Materials submitted for publication are considered contributions to the "pool" of information found in ***TechNotes*** for the advantage of our readers. Although, we do not pay or in any other way compensate you for your contribution, your name shall appear as author or contributor.

Thank you for reading ***TechNotes***. If you do not have a submission for us, write us anyway and let us know what you think about our publication. We certainly appreciate any feedback you may have.

# SUBSCRIBE TODAY TO *TechNotes!*

Don't miss a single issue of Ashton-Tate's hottest source of technical know-how! Each monthly issue of ***TechNotes*** provides you with the most current, accurate technical information available from the Software Support Center on all Ashton-Tate and MultiMate products.

Call our toll free number to subscribe:

(800) 331-4164 (In California)
(800) 828-2514 (Residents of other states)
(619) 487-3212 (In Canada)

If you prefer, fill out the form below and return it with payment to:

***TechNotes* Subscription Service**
**P.O. Box 3729**
**Escondido, CA 92025-0929**

— — — — — — — — — — **(CUT HERE)** — — — — — — — — — —

WE ACCEPT CREDIT ORDERS! Call our toll free number for details. Make checks payable to ASHTON-TATE.

**Amount Enclosed:**

__ $50.00

__ $65.00 (Extra Postage Canada and Mexico)

__ $75.00 (Extra Postage other foreign countries)

__ Bill Me

**Form of Payment: (check one)**

__ Check (**U.S. dollars drawn on U.S. bank**)

__ VISA Number _____

__ MasterCard Number _____

Expiration Date _____

**Signature** _____

**Check One:** __ New Subscription __ Renewal

**Name** _____

**Company** _____

**Address** _____

**City** _____ **State/Province** _____

**Zip/Postal Code** _____ **Country** _____

**Which product(s) do you use?** __ **dBASE II** (D2)
__ **dBASE III** (D3)
__ **Framework** (FW)
__ **MultiMate** (MM)

Please allow 30 to 60 days for delivery of your first issue.

1000

# *TechNotes*
# BACK ISSUES ORDER FORM

Maybe your subscription to ***TechNotes*** just began and you want to have every issue. Maybe you don't subscribe, but want to receive last month's issue. Maybe you want to refer to an article in a past issue to help you design that special application.

Whatever the reason, order Back Issues! Add to your library of dBASE II, dBASE III PLUS , and Framework II technical reference material. To order your Back Issue(s), fill out this form, enclose your payment, and return to:

***TechNotes* Back Issues**
**Ashton-Tate Software Support Center**
**Attn: Kathy Smith**
**20101 Hamilton Ave.**
**Torrance, CA 90502-1319**

Name _____

Company _____

Address _____

City _____ State/Province _____ Zip Code _____

Country _____ Phone Number (_____) _____-

All 1985 Back Issues are $10.00 each, when you order them separately. If you purchase all available 1985 issues as a set, the price is $50.00. All 1986 Back Issues are $7.00 each. All prices include postage and tax.

The following issues are currently available. Check those you wish to order:

| | | |
|---|---|---|
| ( ) FEBRUARY 1985 | ( ) OCTOBER 1985 | ( ) FEBRUARY 1986 |
| ( ) MARCH 1985 | ( ) NOVEMBER 1985 | ( ) MARCH 1986 |
| ( ) APRIL 1985 | ( ) DECEMBER 1985 | ( ) APRIL 1986 |
| ( ) MAY 1985 | (includes 1985 | ( ) MAY 1986 |
| ( ) JUNE 1985 | Annual Disk and | ( ) JUNE 1986 |
| ( ) JULY 1985 | 1985 Cumulative | ( ) JULY 1986 |
| ( ) AUGUST 1985 | Index) | ( ) AUGUST 1986 |
| ( ) SEPTEMBER 1985 | ( ) JANUARY 1986 | ( ) SEPTEMBER 1986 |

$_____ TOTAL AMOUNT ENCLOSED

( ) VISA ( ) MasterCard # _____

Expiration Date _____ Signature _____

**ALL ORDERS MUST BE PREPAID. MAKE CHECKS PAYABLE TO ASHTON-TATE (IN U.S. DOLLARS DRAWN ON U.S. BANK). PLEASE ALLOW SIX (6) WEEKS FOR DELIVERY OF YOUR ORDER.**

NOTE: Information contained in ***TechNotes*** is current as of the issue in which it appears. Therefore, some information found in the Back Issues may no longer be valid (such as Update Policies, etc.).

# Notes

# Notes

# Notes

TechNotes
P.O. Box 3729
Escondido, CA 92025-0929

ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. POSTAGE
**PAID**
PERMIT NO. 27
CULVER CITY, CA